Neural Network Optimization

Martin Walter Simmen

PhD

University of Edinburgh

1992

Declaration

I declare that I have composed this thesis myself and that the research reported herein has been conducted by myself unless otherwise indicated.

Much of the content of chapter 2 has already been published in the following paper:

> Simmen, M.W. (1991) Parameter Sensitivity of the Elastic Net Approach to the Traveling Salesman Problem. *Neural Computation* **3**, 363-374.

Other research arising from issues in the efficient implementation of neural networks on parallel computer architectures, though not reported in this thesis, has been published in:

> Simmen, M. (1991) Comments on broadcast algorithms for two-dimensional grids. *Parallel Computing* **17**, 109-112.

Edinburgh, 23 September 1992.

Martin W. Simmen

Dedication

To my parents, for everything.

Abstract

Combinatorial optimization problems arise throughout science, industry, and commerce. The demonstration that analogue neural networks could, in principle, rapidly find near-optimal solutions to such problems — many of which appear computationally intractable — was important both for the novelty of the approach and because these networks are potentially implementable in parallel hardware. However, subsequent research, conducted largely on the travelling salesman problem, revealed problems regarding the original network's parameter sensitivity and tendency to give invalid states. Although this has led to improvements and new network designs which at least partly overcome the above problems, many issues concerning the performance of optimization networks remain unresolved.

This thesis explores how to optimize the performance of two neural networks current in the literature: the elastic net, and the mean field Potts network, both of which are designed for the travelling salesman problem. Analytical methods elucidate issues of parameter sensitivity and enable parameter values to be chosen in a rational manner. Systematic numerical experiments on realistic size problems complement and support the theoretical analyses throughout.

An existing analysis of how the elastic net algorithm may generate invalid solutions is reviewed and extended. A new analysis locates the parameter regime in which the net may converge to a second type of invalid solution. Combining the two analyses yields a prescription for setting the value of a key parameter optimally with respect to avoiding invalid solutions. The elastic net operates by minimizing a computational energy function. Several new forms of dynamics using locally adaptive step-sizes are developed, and shown to increase greatly the efficiency of the minimization process. Analytical work constraining the range of safe adaptation rates is presented.

A new form of dynamics, with a user defined step-size, is introduced for the mean field Potts network. An analysis of the network's critical temperature under these dynamics is given, by generalizing a previous analysis valid for a special case of the dynamics. Understanding the parameter dependence of this temperature clarifies an earlier problem in the use of synchronous updating dynamics, and is vital in choosing values for the network's free parameters. A variety of mathematical methods are developed to set the optimal values of the two free coefficients in the system's energy function . Finally, the experimental solution quality for Euclidean and non-Euclidean problems is contrasted, and an explanation, using arguments about the structure of the free energy landscape, proposed to account for the poorer performance on non-Euclidean problems.

Acknowledgements

I have a great deal to thank my supervisors for. David Wallace's guidance and incredible enthusiasm were both invaluable. His faith in my work, despite my occasional academic meanderings, was very important to me. David Willshaw has been a source of sound advice and encouragement throughout this project. Particularly during the past few months, his good humour, tolerance, and strong coffee have been very much appreciated. Thank you both.

My Rm.4408 office-mates, Hon Yau and Jonathan Pryce, have been subjected to many naïve questions about statistical mechanics, yet they have both become good friends, making life in the JCMB far more tolerable than it would otherwise have been. Hon made countless invaluable comments and criticisms on a final draft of this thesis, his heroic efforts going far beyond the call of duty. Jonathan's advice, dry wit, tales from the 'world outside', and willingness to act as a grammatical spot-checker, were most welcome. I wish also to thank Karl Bedlow, ex of 4408, for many stimulating discussions about physics and Life.

I also want to thank Peter Dayan and Marcus Frean for contributing to my understanding of the elastic net and, more importantly, broadening my interests. For lively discussions about parallel computing and the politics of science, I thank Tom Tollenaere.

Frank Smieja, Jeremy Craven, Alan Simpson, and most recently Toby Tyrrell have helped by providing much needed sporting diversions.

I also wish to express my gratitude to the many people in the Edinburgh Parallel Computing Centre with whom I have dealt. Although I frequently grumbled, they provided me with access to computer facilities unrivalled elsewhere in the UK.

Finally, I reserve my deepest thanks for Shari Cohn. Her constant encouragement, friendship and love over the years have been inspirational.

The Science and Engineering Research Council provided financial support for which I am grateful.

Contents

1 Combinatorial Optimization **1**

1.1 Introduction 1

1.2 Combinatorial Problems and Applications 5

1.3 $\mathcal{NP}$-Complete and $\mathcal{NP}$-Hard Problems 9

1.4 Conventional Methods 12

1.5 Novel Methods 16

1.6 Theoretical Results on the TSP 26

2 Parameter Sensitivity of the Elastic Net Algorithm **29**

2.1 Introduction 30

2.2 A Brief Review of Previous Elastic Net Work 31

2.3 Sensitivity to the Value of β/α 32

2.4 Stable Non-Tour Configurations in the $K \to 0$ Limit 34

2.5 Simulations and Discussion 41

2.6 Conclusions 46

3 Enhanced Minimization Methods for the Elastic Net **48**

3.1 Introduction 49

3.2 Energy and Momentum Methods 52
3.3 Local Step-Size Adaptation Methods 56
3.4 Performance on Random Euclidean TSPs 65
3.5 Conclusions 72

4 Potts TSP Network Dynamics **73**
4.1 Formalism 73
4.2 Parallel Updating $T_c(\beta, \gamma)$ Analysis 81
4.3 Serial Updating $T_c(\beta, \gamma)$ Analysis 87
4.4 Numerical Tests of the $T_c(\beta, \gamma)$ Analysis 89
4.5 Conclusions 100

5 Potts TSP Network: Parameter Optimization and Performance **101**
5.1 Choosing the Value of Coefficient α 102
5.2 Choosing the Value of Coefficient β 108
5.3 Performance 121
5.4 Conclusions 127

6 Conclusions **129**

A Parallel Implementation Strategies **131**

B Theory For Random Metric TSPs **135**

Bibliography **138**

List of Figures

1.1 Example of a graph bisection problem 6

2.1 Sketch of the spike problem in the elastic net 33

2.2 Algorithm failure due to a 'frozen bead' 35

2.3 Sketch of 'frozen bead' development in the elastic net 36

2.4 Defect frequency plotted against γ/γ' for the elastic net 42

2.5 Tour quality plotted against γ/γ' for the elastic net 45

3.1 Sketch illustrating elastic net high K stability constraint 51

3.2 Sketch illustrating elastic net low K stability constraints 62

3.3 Tour quality *vs.* cooling rate plots for lattice TSPs 64

3.4 Tour quality *vs.* cooling rate plots for random TSPs 67

3.5 Comparison of the best Or*-opt and elastic net 50 city tours . . . 69

4.1 Parallel dynamics T_c vs. T_o plots for various (ξ, Δ); Potts method 93

4.2 Parallel dynamics T_c vs. T_o plots for various K_r; Potts method . 94

4.3 Tour quality vs. T_c for various (ξ, T_r) in the Potts method 96

4.4 Experimental and theoretical parallel dynamics $T_c(\beta)$ data . . . 97

4.5 Experimental and theoretical parallel dynamics $T_c(\gamma)$ data . . . 98

5.1 Plots of the restraint energy E_α vs. α in the Potts method 107

5.2 Plots of the saturation vs. β in the Potts method 116

5.3 Experimental and theoretical low β data in the Potts method . . 117

List of Tables

3.1 Tour-length data using the steepest descent and energy monitoring techniques in the elastic net algorithm 53

3.2 Tour-length data using the momentum technique 55

3.3 Tour-length data using locally adaptive step-size techniques . . 58

5.1 Some example values of β^* calculated from equation 5.16 114

5.2 Tour-length data for 50 city Euclidean TSPs; Potts method 123

5.3 Tour-length data for 200 city Euclidean TSPs; Potts method . . . 123

5.4 Tour-length data for random metric TSPs ($N = 50, 100, 200$); Potts method 125

CHAPTER 1

Combinatorial Optimization

1.1 Introduction

Unlike many topics in modern theoretical physics, that of optimization is something with which we are all familiar in our everyday lives, be it through finding the shortest route home from the office, or arranging a diary to accommodate competing demands on our time each day. A vast number of such problems, in which the task is to find the optimal solution amongst a space of possible solutions subject to external constraints on that choice, arise throughout science, engineering, and industry. Problems in which the possible solutions form a finite set are termed *combinatorial optimization* problems. For this class of problems the standard function max(min)imization techniques of differential calculus are clearly not applicable. Since the 1940s however, new developments in mathematics e.g. linear programming and advances in graph theory, have led to a large body of knowledge — termed *operations research* — concerning these problems and methods for their exact or approximate solution.

The need for efficient approximate solution techniques remains acute, for two main reasons. Firstly, in real-world optimization problems (e.g. as arise in VLSI

design) the number of variables N may be as high as 10^6, leading to enormous search spaces. Secondly, research in computational complexity theory during the 1970s revealed a set of equally 'hard' problems, the '$\mathcal{NP}$-complete' set, for which it is unlikely that any algorithm guaranteeing the exact solution and running in a time polynomial in N, will exist. As many of the most important combinatorial optimization problems lie within this set, much effort has been directed at developing heuristic methods which settle for finding near-optimal solutions but run in low order polynomial time.

During the last decade, it was realised — by regarding solution costs as energies — that the solution spaces of large combinatorial optimization problems often share similarities with the configuration spaces in the spin glass models of disordered systems so intensively studied in statistical mechanics. This remarkable connection, between two such apparently disparate fields, has led both to fresh insights into the nature of the optimization problems themselves and to two radically new approaches for finding approximate solutions, *viz*, simulated annealing and neural networks.

Briefly, simulated annealing searches for the optimal solution (the ground state in physics terminology) through Monte Carlo thermalization at decreasing values of an appropriately defined 'temperature', on the grounds that at low temperature the Boltzmann distribution will select out the lowest cost solutions. The neural network methods differ from most of the operations research methods and simulated annealing, in that whereas the latter approaches are restricted to searching within the finite set of feasible solutions, the neural methods explore a continuous search space *en route* to picking out a solution. Another distinctive feature of the neural methods is that most of them are inherently parallel in operation. Thus they are ideally placed to exploit the trend towards parallel architectures in current high performance computing. Some of them are also potentially implementable in hardware.

Given their theoretical novelty and potential benefits, it is no surprise that the performance and behaviour of neural optimization algorithms have been, and continue to be, topics of active research. There has, however, been a slight tendency for researchers each to invent their own algorithms rather than to explore fully aspects of existing algorithms. This thesis presents a study of two neural network algorithms current in the literature, both of which are designed to find approximate solutions to the *travelling salesman problem* — the most widely studied combinatorial optimization problem. The two algorithms are the *elastic net* [Durbin & Willshaw 1987] and the *mean field Potts* algorithm [Peterson & Söderberg 1989]. A combination of analytical techniques and numerical simulations is used to investigate three key issues in these algorithms:

- parameter sensitivity and optimal parameter choice
- choice of suitable dynamics
- limitations on performance

All of the simulations were performed on powerful parallel computer systems, which facilitated extensive experimental studies on non-trivial sized problems. Progress has been made with respect to each of the issues above, though, inevitably, some open questions remain.

Layout of the Thesis

Chapter 2 defines the elastic net algorithm and studies the sensitivity of its performance to the ratio of its two key free parameters, looking in particular at how this ratio affects the likelihood of convergence to invalid solutions. Stability analyses correctly account for the observed simulation trends.

Chapter 3 focuses on optimizing a second aspect of the elastic net algorithm, namely the way its dynamics locate energy minima. Self-adaptive dynamics are developed and shown to increase greatly the efficiency of the algorithm. Analytical work constraining the range of robust adaptation rates is presented and verified experimentally.

Chapter 4 studies the parameter dependence of an important characteristic temperature in the mean field Potts network for the travelling salesman problem. An existing analysis for iterative dynamics is generalized to the case of dynamics with an arbitrary step-size, γ. This reveals a critical value of γ, above which the algorithm fails to have the correct scaling behaviour for large problems when using synchronous dynamics. In testing the theory against experimental data, the sensitivity of the network's behaviour to various secondary factors is demonstrated and discussed.

Finally, chapter 5 studies further issues in the mean field Potts network. A variety of mathematical methods to set the optimal values of the two coefficients in the system's energy function are developed, and shown to be in decent accord with simulation data. The quality of the tours found for Euclidean and non-Euclidean problems is then investigated and contrasted, for problems with up to 200 cities. An explanation is proposed to account for the poorer performance on non-Euclidean travelling salesman problems.

The remainder of this chapter provides the context and background information for the four research chapters just outlined. Section 1.2 defines some important combinatorial optimization problems and attempts to motivate their study with reference to applications. The important computational complexity results are then summarized in section 1.3. Section 1.4 briefly surveys the solution methods developed from operations research, concentrating on the travelling salesman problem (TSP). Section 1.5 surveys the novel methods, focusing naturally on the neural network TSP approaches, and the final section summarizes relevant

theoretical results concerning the TSP.

1.2 Combinatorial Problems and Applications

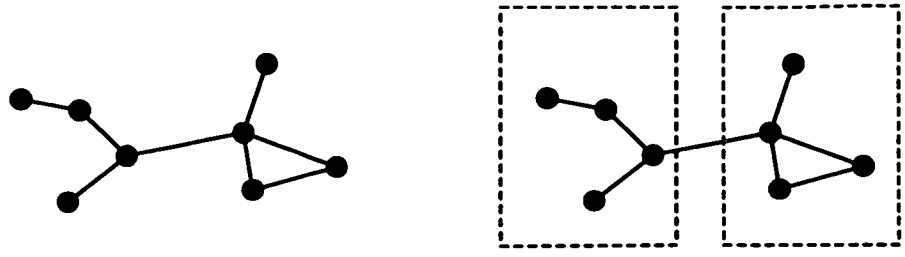
This section gives a generic definition of combinatorial optimization problems, followed by three specific examples, with a brief mention of applications for each. The three problems are graph partitioning (GPP), the travelling salesman problem (TSP), and the knapsack problem (KP). The purpose is to place the later work, which focuses exclusively on the TSP, into a wider context, but no attempt will be made to survey the full range of combinatorial problems — for that the reader can refer to [Garey & Johnson 1979], and references therein.

There is no absolutely agreed definition of what constitutes a combinatorial optimization problem, but the following statements (after [Garey & Johnson 1979]) capture the essential elements. Without loss of generality they refer to minimization problems.

DEFINITION
A combinatorial optimization problem has three properties:

1. a set $\mathcal{D}$ of instances, each of which can be considered as input data.
2. for each instance $\phi \in \mathcal{D}$, there is a finite set S of possible solutions.
3. a mapping which, for each instance ϕ and $\omega \in S$, assigns a positive real number $C(\omega)$, the *cost* of solution ω.

The task is then to find, for instance ϕ, the solution(s) of minimal cost, such solution(s) being referred to as *optimal* or *exact* (and having cost C_{opt}).



(a) (b)

Figure 1.1: Graph Bisection example. The graph is displayed in part (a) and the optimal bisection, with cut-size 1, is shown in (b). All edge weights equal.

Algorithms which guarantee the exact solution for every instance are termed *optimization algorithms*, the rest are deemed *approximation algorithms*.

1.2.1 The Graph Partitioning Problem

Consider an undirected graph $G = (V, E)$, where V is the set of N vertices and each edge $e \in E$ has an associated (positive integer) weight $l(e)$. Given some integer $K \geq 2$ which is a divisor of N, the problem is to partition V into K disjoint sets $V_1 \cdots V_K$ of equal cardinality such that, letting $E' \subseteq E$ denote the set of edges which connect vertices in different sets V_i, the cut-size $\sum_{e \in E'} l(e)$ is minimal.

Figure 1.1 shows a simple example for $K = 2$. Basic combinatorics leads to the number of distinct partitions (W) being

$$W = \frac{1}{K!} \prod_{j=0}^{K-2} C_m^{N-jm}, \quad \text{where } m \stackrel{def}{=} \frac{N}{K}, \; C_k^l \stackrel{def}{=} \frac{l!}{k!(l-k)!}. \tag{1.1}$$

For the graph bisection case ($K = 2$), this implies that as $N \to \infty, W \to 2^{N-1}$. Thus for large graphs the number of possible solutions becomes exponentially large.

Graph partitioning-like problems often arise in VLSI design. For example, if the number of components in a circuit is such that they must be split over several chips, then it is imperative to keep regions of the circuit with a high interconnection density within a chip, so as to minimize the number of expensive chip-to-chip wires. Such problems can clearly be mapped onto the GPP. In practice, it is often desirable to allow slight violations of the equipartition constraint; this can be dealt with by defining a new cost function comprising the cut-size term plus an equipartition violation term proportional to $(\sum_{i=1}^{K} |V_i|^2 - Km^2)$.

1.2.2 The Travelling Salesman Problem

The TSP is the problem of finding the shortest closed path around a number of points. Formally, each instance is specified by a set of N points (the 'cities') labelled $i = 1 \ldots N$, and an $N \times N$ matrix of positive inter-city 'distances' d_{ij}. Each tour round the cities can be represented by a permutation π of the N points, with the tour-length given by

$$\mathcal{L}_\pi = \sum_{i=1}^{N} d_{\pi(i)\pi(i+1)} \tag{1.2}$$

where $\pi(N+1) = \pi(1)$. The task is to find the tour with the minimal $\mathcal{L}$.

For symmetric TSPs, i.e. those where $d_{ij} = d_{ji} \, \forall \, i, j$, there are $\frac{1}{2}(N-1)!$ distinct tours, indicating that an exhaustive search strategy is untenable for large N. The symmetric class can usefully be broken down into the Euclidean and non-Euclidean subclasses. In the former, cities are specified by coordinates and the d_{ij} are computed in a Euclidean metric, whereas in the latter class each instance is specified purely by a matrix of d_{ij} terms.

A large number of practical optimization problems can be framed as TSPs, either directly or indirectly. The 2-D Euclidean TSP, for example, describes a goods distribution problem in which a truck must deliver items to a number

of locations and then return to the depot, using the shortest route. Similarly, the question of optimizing the route of a drill which must produce thousands of holes in a circuit board can also been seen as a 2-D geometrical TSP. On the other hand, simple scheduling problems can be represented by non-Euclidean TSPs. Consider for example the problem of scheduling N jobs, $i = 1 \ldots N$, to run sequentially on a single machine in the shortest time. Job i requires time t_i, plus a lag-time d_{ji} in which the machine is altered to allow execution of job i after the previous job j. The initial (and final) state can be regarded as a dummy job, indexed by 0, with $t_0 = 0$. Representing a schedule by a permutation π of jobs $0 \ldots N$, with $\pi(N+1) = \pi(0)$, the total run time T equals

$$\begin{aligned} T &= \sum_{i=0}^{N} (d_{\pi(i)\pi(i+1)} + t_{\pi(i)}) \\ &= \sum_{i=0}^{N} d_{\pi(i)\pi(i+1)} + \sum_{i=0}^{N} t_i \end{aligned}$$

As $\sum t_i$ is independent of π, minimizing T is equivalent to an $(N+1)$-city non-Euclidean TSP (which may be either symmetric or asymmetric).

Further applications, extensions and variations of the basic TSP formulation can be found in the standard text on the subject [Lawler *et al.* 1985].

1.2.3 The Knapsack Problem

Given a set of objects, each with a certain size and utility, and a knapsack of finite size, the problem arises as to how to maximise the utility of objects placed in the knapsack. Formally, consider N types of object labelled $i = 1 \ldots N$. Let s_i and u_i denote the size and utility respectively of type i objects, and S the knapsack size, with $S, s_i, u_i \in Z^+ \forall i$. The problem is to find the set of integers x_i such that $\sum_i x_i u_i$ is maximised, subject to the size constraint $\sum_i x_i s_i \leq S$.

As in the GPP and TSP, the configuration space explodes for large N. For example, in the special case in which the x_i are also constrained by $0 \leq x_i \leq K-1$, $\forall i$,

the total number of configurations is clearly K^N, though in general not all of these will satisfy the global size constraint.

As a general example of constrained resource utilisation, the knapsack problem (KP) has found a wide range of applications throughout industry and business.

1.3 $\mathcal{NP}$-Complete and $\mathcal{NP}$-Hard Problems

To date, no optimization algorithm running in polynomial time is known for any of the three problems just discussed. Furthermore, in light of some remarkable computational complexity theorems due to Cook and others in the 1970s, it is widely believed that such algorithms are unlikely to exist. This section will cover only the main points of this important topic; for further details and history consult [Garey & Johnson 1979] and references therein.

The starting point is the idea that problems can be regarded as either 'easy' or 'intractable', depending on whether they can always (i.e. for every instance) be solved by an algorithm with running time polynomial in the input data size. For technical reasons, the theory explores this classification scheme primarily for 'decision' problems, i.e. problems with Yes/No answers, rather than optimization problems. However, an optimization problem Ψ often has a decision version Ψ_D (e.g. TSP_D asks whether there exists a tour of length $\leq L$), and this allows optimization results to be derived too. This will become clearer as we proceed.

Let $\mathcal{P}$ denote the set of easy decision problems. Many problems are known to lie in $\mathcal{P}$, the assignment and minimal spanning tree decision problems being just two examples. There are also problems proven to be in $\mathcal{P}'$: these being either undecidable (i.e. incapable of being solved by any algorithm), or, decidable but

requiring run times worse than exponential for some instances. Then there is a third class of problems, those whose status with respect to $\mathcal{P}$ is currently uncertain. Such problems are apparently intractable, though this has never been proven. To filter out these interesting problems from those which are provably intractable, the abstract (and unphysical) notion of a *nondeterministic algorithm* was introduced. Roughly speaking, such algorithms can branch off several parallel decision processes and execute these concurrently. If the algorithm can always produce a correct response to every 'yes' instance of Ψ_D within a polynomially bounded number of levels in the tree, then Ψ_D is said to be solvable in nondeterministic polynomial time, and is a member of the set $\mathcal{NP}$.

Although it is clear that $\mathcal{P} \subseteq \mathcal{NP}$, the issue of whether $\mathcal{P} \neq \mathcal{NP}$ remains open. This question can be cast into sharper relief by consideration of the '$\mathcal{NP}$-complete' set ($\mathcal{NPC}$) — the subset of $\mathcal{NP}$ containing the hardest problems. Formally, $\Psi_D \in \mathcal{NPC}$, if i) $\Psi_D \in \mathcal{NP}$, and, ii) every $\Psi'_D \in \mathcal{NP}$ is polynomially transformable to Ψ_D. Polynomial transformability means that Ψ'_D can be solved by an algorithm which calls up a Ψ_D 'subroutine' once, and whose running time, excluding this call, is polynomially bounded.

Many hundreds of important decision problems, including TSP_D, GPP_D and KP_D, have now been proven to be $\mathcal{NP}$-complete. The importance of the $\mathcal{NP}$-complete set is that, by construction

$$\exists \Psi_D \in \mathcal{NPC} : \Psi_D \in \mathcal{P} \Longleftrightarrow \mathcal{P} = \mathcal{NP}. \tag{1.3}$$

Thus the failure so far to find a polynomial time algorithm for any $\mathcal{NP}$-complete problem, despite strenuous efforts, is the main reason for the folk-belief that $\mathcal{P} \neq \mathcal{NP}$ and hence that all $\mathcal{NP}$-complete problems are inherently intractable.

These results can be applied to optimization problems through the concept of the '$\mathcal{NP}$-hard' class, containing all the problems "at least as hard as those in $\mathcal{NPC}$." Formally, a problem is $\mathcal{NP}$-hard if there is an $\mathcal{NP}$-complete problem which is

polynomial time reducible to it.[1] Many optimization problems can be proven $\mathcal{NP}$-hard by consideration of the polynomial reducibility of the corresponding decision problem. By construction, $\mathcal{P} = \mathcal{NP}$ is a necessary condition for any $\mathcal{NP}$-hard problem to be solvable in polynomial time. Furthermore, for TSP, GPP, KP and many other optimization problems Ψ, it can also be proven that Ψ_D is polynomially reducible to Ψ, and hence that these problems are of equivalent computational complexity to their decision versions.

Polynomial-time Approximation Schemes

In the expectation that $\mathcal{P} \neq \mathcal{NP}$, it is tempting to ask whether, in return for abandoning the goal of true optimization algorithms, one can devise methods for $\mathcal{NP}$-hard problems which guarantee solutions within a certain range of the optimum, and which run in polynomial time. Specifically, for an optimization problem Ψ, is there an algorithm A such that for some arbitrary input parameter $\epsilon > 0$, the cost of the solution found by A on each input instance ϕ, C_A^ϕ, satisfies

$$\frac{C_A^\phi - C_{opt}^\phi}{C_{opt}^\phi} \leq \epsilon. \tag{1.4}$$

and for which the running time is polynomial in both the problem size and $1/\epsilon$. A 'fully polynomial-time approximation scheme' such as this is in fact known for the knapsack problem. Thus KP, though $\mathcal{NP}$-hard, is "well-behaved" from the viewpoint of finding approximate solutions. Alas the same cannot be said of the TSP and many other $\mathcal{NP}$-hard problems. Even for TSPs in which the distances obey the triangle inequality, the best-known performance guarantee — from the Christofides' algorithm [Lawler *et al.* 1985, chapter 5] — only gives a fixed ϵ, value 1.5, with $O(N^3)$ running time. Furthermore, a fully polynomial-time approximation scheme for the TSP is unlikely in principle, since it has been proven (in the above reference) that its existence would imply $\mathcal{P} = \mathcal{NP}$.

[1] Where polynomial reducibility is a relaxation of polynomial transformability, allowing a polynomial number of subroutine calls.

Hence the interest in TSP heuristics which, even if they carry no (or very weak) worst-case guarantees, might be more fruitful paths to approximate solutions.

1.4 Conventional Methods

Despite the ostensibly negative news from the computational complexity front regarding the TSP, a variety of approaches has been developed which usually offer a trade-off between solution quality and running time for practical problems. This reflects the fact that the complexity results just discussed are *worst-case* results, which may, of course, be unrelated to the average-case behaviour. This section briefly sketches the three main types of approach commonly cited in the operations research literature. Although these comments specifically concern the TSP, the solution methods for many other combinatorial optimization problems have a similar taxonomy.

1.4.1 Integer Programming Methods

These are powerful, highly sophisticated techniques dedicated to searching for optimal or very nearly optimal solutions, at the expense of long run times and complex software requiring thousands of lines of code. Their starting point is a tour representation in terms of bond variables x_{ij}, where (for symmetric TSPs) $x_{ij} = 1$ if cities i and j are adjacent in the tour, and is 0 otherwise. The problem is to minimize

$$\sum_i \sum_{j>i} x_{ij} d_{ij} \tag{1.5}$$

subject to constraints

$$\sum_{j>i} x_{ij} + \sum_{j<i} x_{ji} = 2 \qquad \forall i \tag{1.6}$$

$$\sum_{i \in U} \sum_{j \in U, j > i} x_{ij} \leq |U| - 1 \qquad \forall U \subset V, U \neq \emptyset \tag{1.7}$$

where V is the set of city vertices and $|A|$ denotes the cardinality of set A. The first constraint requires each city to be connected to two others, and the second ensures that subtours — in which a group of less than N cities are connected in a closed loop — are prevented. The $\{x_{ij}\}$ sets which satisfy the constraints can be viewed as points in an $O(N^2)$-dimensional space. If a sufficient number of constraint equations are used, a polyhedron can be specified in this space, the vertices of which correspond to tours. Linear programming methods could then be used, in principle, to determine the optimal tour. Unfortunately there are an exponential number of the subtour elimination constraints (1.7), so a smaller number of more powerful inequalities are sought instead, through exploiting the relationship between the TSP and some tractable optimization problems. Such 'polyhedral' techniques are often combined with 'branch and bound' procedures, which essentially prune the search space in an intelligent fashion; a review of these techniques can be found in [Lawler *et al.* 1985, chapters 8-10].

Although not guaranteed to work on every TSP instance, these methods typically find very high quality solutions, and furthermore, unlike any of the other methods discussed later, are often capable of proving the optimality of their solutions. Experiments reviewed in the reference above found the optimal solutions for several symmetric 100 city problems in only a few minutes of supercomputer time, and the optimal solution of a 2392 city instance in only a few hours has been reported in [Padberg & Rinaldi 1988]. Methods for the asymmetric TSP are less advanced, but for some distance distributions large instances can still be solved to optimality [Miller & Pekny 1991].

1.4.2 Tour Construction Heuristics

In contrast to polyhedral methods, tour construction heuristics are usually transparent, simple to program, and generate tours in low order polynomial time. It comes as no surprise therefore that their performance is almost invariably far from optimal. As the name suggests, these heuristics start with a structure which is not currently a tour e.g. a single city, a set of edges or a cycle, and then sequentially add edges (according to some rule) until a tour is constructed. The 'nearest neighbour' heuristic, for example, starting at some arbitrary city, builds up a path by adding, at each step, the (as yet unselected) city nearest the city currently at the end of the path. After $N-1$ such steps the path is closed by returning to the initial city. The empirical performance of this and ten other tour construction heuristics was investigated in [Johnson 1990]. With respect to optimal tour-length lower bounds derived using the reputable Held-Karp method [Held & Karp 1970], the heuristics produced tours ranging from 10% to 60% longer than the bound, on uniformly random Euclidean TSPs with $N = 10^2$–10^5. Nearest neighbour, for example, typically runs up a tour-length percentage excess of between 20 and 30 across this range of problems.

Tour construction heuristics have arguably three uses:

- the quality-level of their tours serves as a threshold which any new TSP algorithms, and neural network approaches specifically, must cross before they can expect to receive any credibility.
- for extremely large problems in which running time minimization is important and a large degree of sub-optimality in the tour is acceptable, as is sometimes the case in engineering applications.
- as sources of initial tours for local optimization methods.

1.4.3 Local Optimization Heuristics

These algorithms take some existing solution and search over a set of 'neighbouring' solutions, looking for solutions of lower cost. If one is found, then the algorithm adopts it as the new state, and performs another neighbourhood search centred on it. This procedure is repeated until no further improvements are found i.e. until the solution is a local minimum with respect to the particular neighbourhood structure used. For the TSP, the most successful neighbourhoods are those defined in terms of *r-edge exchange* operations, in which a tour's neighbours are those tours which could be formed from it by deleting r edges in the tour and exchanging them with r edges currently not in the tour, subject to the result still being a valid tour. Local minima with respect to r-edge exchange are termed 'r-opt' tours.

The first serious study of these methods was undertaken by [Lin 1965], who investigated the 2-Opt and 3-Opt algorithms. This led to the development of the celebrated Lin-Kernighan algorithm (LK) [Lin & Kernighan 1973], in which the fixed r search idea is extended to yield a powerful variable depth partial search technique. LK and 3-Opt are very widely used, as they typically give good tours in acceptable time and, for 3-Opt at least, can be implemented relatively easily. [Johnson 1990] reports empirical values for the excess over the Held-Karp lower bound, on uniformly random Euclidean instances with $N = 10^2$–10^5, of approximately 6%, 3%, and 2%, for 2-Opt, 3-Opt and LK respectively. Empirically, all three heuristics appear to run in $O(N^2)$ time. Note however that this result has no theoretical underpinning: although the neighbourhood search time is polynomially bounded for low r (as the neighbourhood set is of size $O(N^r)$ for r-edge exchange), there is no guarantee that the number of searches needed to reach a local optimum is also polynomially bounded.

The Or-opt algorithm utilises a restricted 3-opt search to find good tours rapidly [Lawler *et al.* 1985]. The neighbourhood is defined by the 'k-city insertion'

operation, in which an edge (i, j) in the tour is deleted and a randomly selected segment of k cities currently elsewhere in the tour is inserted between cities i and j (and closing up the tour at the point of segment excision). Or-opt finds a local optimum with respect to three city insertion, then uses this as the initial state for two city insertion optimization, followed by a similar process using a single city. A variant of this method, Or*-opt, was developed and used in the current study to provide benchmark data for the neural network approaches. Or*-opt generalizes this to the case $k_o = N - 2$ rather than 3, and furthermore repeats the whole sequence of k-city insertion optimizations until the tour is locally optimal for all values of k.

1.5 Novel Methods

During the last decade several radically new approaches to tackling optimization problems have emerged. They are novel in that they are all inspired by optimization principles active in natural systems. Of the three approaches — genetic algorithms, simulated annealing, and neural networks — we shall cover only the key points of the first two, before concentrating on the neural algorithms. Whilst there are (to the author's knowledge) as yet no neural TSP algorithms incorporating concepts from genetic algorithms (GAs), the latter have recently been combined with both local optimization and simulated annealing to good effect, and so a few comments on GAs are warranted.

1.5.1 Genetic Algorithms

Genetic algorithms are attempts to abstract and utilise the key mechanisms operating in the evolution of biological systems. Through natural selection, nature has developed a robust and powerful technique for maximising the

'fitness' of populations. GAs mimic this by evolving a population of solutions, generating new solutions by a 'mating' procedure in which parents are chosen probabilistically according to their fitness, i.e. solution cost. The offspring incorporate elements of their parents' solutions, and so after many generations the solutions in the population tend to improve. To maintain diversity in the population, a low level of random 'mutation' is also incorporated. (See [Goldberg 1989] for a proper introduction to the subject.)

Although there have been some TSP studies using the pure GA method, arguably better results have been obtained using hybrid GA systems incorporating r-opt local search moves [Brady 1985, Mühlenbein *et al.* 1988]. Not surprisingly, such systems can perform as well as local optimization methods, indeed the technique of [Mühlenbein *et al.* 1988] has found tours within only a few percent of the optimal length, for large instances. In other research, the concept of an interacting population of solutions has been wedded to the simulated annealing technique to produce an interesting TSP algorithm [Ruján 1988], and this concept may have wider applicability in search problems.

1.5.2 Simulated Annealing

In computational experiments studying the equilibrium statistical mechanics of many-particle systems, the Metropolis algorithm is used as a standard method for generating the Boltzmann distribution of microstates characteristic of thermal equilibrium. [Kirkpatrick *et al.* 1983] proposed that the Metropolis algorithm could also be applied to combinatorial optimization problems, by viewing the set of possible solutions as the microstates, solution cost replacing energy, and introducing an appropriately defined 'temperature' T. Lowering T towards zero in a sequence of steps, whilst ensuring thermal equilibrium is maintained throughout, the system will increasingly tend to sample configurations of low cost, in accordance with the Boltzmann distribution sharpening

around the ground state as $T \rightarrow 0$. The name of the new method is a reference to the similar process of slowly cooling a melt so that the resulting crystalline structure is of low energy, with few random defects.

A comprehensive discussion of the theoretical and practical details of simulated annealing is given in [van Laarhoven & Aarts 1987]. It suffices here to say that the elementary operation in the Metropolis method for the TSP is the generation of some new candidate configuration (usually via a 2-opt or 3-opt move), which is then automatically accepted if it lowers the solution cost, or is accepted with probability $\exp(-\Delta C/T)$ if it would increase the cost by ΔC. Thus the technique is essentially a generalization of the local optimization strategy, where, at non-zero temperatures, thermal excitations can facilitate escape from local minima.

Simulated annealing is a robust technique and has been applied with success both to standard combinatorial optimization problems and to some 'messy' problems in VLSI design. However, to obtain good results the annealing runs often require a great deal of computer time. [Johnson *et al.* 1989] criticised much of the earlier research for only comparing simulated annealing with the simpler local optimization heuristics (i.e. those using r-opt style moves) and not with the more competitive LK-style heuristics. They undertook a systematic comparison of the above three methods' performance on the graph bisection problem, for low degree random and random geometric graphs. When assessed using equivalent amounts of computer time, simulated annealing emerged a clear winner over simple local optimization for both types of graph. Against the LK-style algorithm however, the new technique was only slightly better for random graphs and worse for the random geometrical class. A similar investigation is currently underway for the TSP [Johnson *et al.* 1992]; preliminary results [Johnson 1990] suggest that, matched for run-time, simulated annealing beats 3-Opt but is worse than LK.

In summary, simulated annealing appears competitive with, but probably

no better than, the best local optimization methods. Its major drawback is the requirement for long run times. Furthermore, the inherently serial nature of the approach in general precludes any obvious parallel implementation, though several attempts have been made to overcome this problem [van Laarhoven & Aarts 1987, chapter 8].

1.5.3 Neural Network Algorithms

These algorithms draw their inspiration from idealised models of the dynamical behaviour of richly interconnected networks of nonlinear neurons. Here, and throughout this thesis, the emphasis will be on the mathematical rather than biological aspects of these networks. Much of the work in this area has centred on understanding and improving on the Hopfield-Tank TSP algorithm (HT) [Hopfield & Tank 1985]. We shall first examine HT and its problems, then introduce the Potts style method which largely overcomes these problems, and finally discuss topographic mapping TSP algorithms.

The Hopfield-Tank TSP Algorithm

In two seminal papers, Hopfield first investigated the asynchronous dynamics of an interconnected network of discrete two-state 'neurons' [Hopfield 1982] and then extended this study to nonlinear analogue neurons with continuous-time dynamics [Hopfield 1984]. Under certain conditions on the connectivity matrix T, it was shown that in each model the local computations of the neurons collectively act to minimize some global 'energy' function. Specifically, for the analogue case, the evolution of the network is governed by a set of nonlinear coupled differential equations

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -u_i + \sum_{j=1}^{n} \mathrm{T}_{ij} V_j + I_i \tag{1.8}$$

where $V_j \in [0,1]$, the output of neuron j, is computed as a monotonically increasing function $g(u_j)$ of its input (or 'potential') u_j. The sigmoid function $g_\lambda(x) = (1 + e^{-2\lambda x})^{-1}$ is conventionally used as the input-output function. Notice that for large λ (i.e. high 'gain') g_λ approximates a step function, forcing the V_i to tend to either 0 or 1.

Hopfield proved that for symmetric T the above dynamics have an associated Lyapunov function

$$E' = \left(-\frac{1}{2} \sum_{ij} \mathrm{T}_{ij} V_i V_j - \sum_i I_i V_i \right) + \frac{1}{\lambda} \sum_i \int_0^{V_i} g_{\lambda=1}^{-1}(V) \mathrm{d}V. \tag{1.9}$$

Denoting the network state by $\boldsymbol{V} \equiv (V_1 \ldots V_n)$, we therefore have an energy function defined throughout $\boldsymbol{V} \in [0,1]^n$ and dynamics which guarantee convergence to minima. For suitable T, the minima of the first term lie at hypercube vertices. The second term is minimized at the hypercube centre but has negligible impact in the high λ regime because of the $1/\lambda$ prefactor. For other values of λ it acts to pull the E' minima slightly away from the vertices.

[Hopfield & Tank 1985] proposed that combinatorial optimization problems could be solved using such analogue networks. To achieve this, one needs a representation of the problem in which the feasible solutions lie at hypercube vertices, and a quadratic energy function (to be mapped onto term 1 of (1.9)) for which the minima correspond to solutions and the depth of each minimum reflects the solution quality. Aside from its intellectual interest, this approach also offers potentially great practical benefits, as (1.8) can also describe the behaviour of a network of interconnected electrical amplifiers, and hence hardware cicuits might be capable of solving hard optimization problems in real time. In practice however, even at a software level of implementation, considerable difficulties have been encountered carrying this program out for the TSP.

HT uses a permutation matrix representation of TSP tours. In this representation, $V_{ia} = 1/0$ means that city i is/(is not) the a th city visited in the tour.

Regarding i and a as row and column indices respectively, tours are represented by $[V_{ia}]$ matrices with exactly one 1 in each row and column, i.e. they are permutation matrices.

The energy function (below) consists of four terms. The first three are restraint terms which penalize all states which are not tours; their joint sum is zero if and only if the system is at a vertex representing a tour. Only the last term contains instance-specific information; given that the restraint term is zero, it gives the length of the tour (where $a \pm 1$ is evaluated modulo N).

$$\begin{aligned} E &= \frac{A}{2}\sum_i\sum_a\sum_{b\neq a} V_{ia}V_{ib} + \frac{B}{2}\sum_a\sum_i\sum_{j\neq i} V_{ia}V_{ja} + \frac{C}{2}(\sum_{ia} V_{ia} - N)^2 \\ &\quad + \frac{D}{2}\sum_i\sum_j\sum_a d_{ij}V_{ia}(V_{j,a+1} + V_{j,a-1}) \end{aligned} \tag{1.10}$$

Mapping E onto the bracketed term in (1.9) yields a prescription for the T_{ij} and I_i terms to be used in the dynamics (1.8). That leaves values for the parameters A, B, C, D, and λ plus the initial state to be specified. Taking an unbiased initial state $V_{ia} \approx 1/N \ \forall i, a$ and setting $B = A$ were two obvious steps taken by Hopfield and Tank. After setting the other parameters by "anecdotal exploration" they reported that, using multiple runs with different initialization noise, 80% of the runs converged to tour states for an $N = 10$ Euclidean TSP. Other researchers, notably [Wilson & Pawley 1988], found far lower levels of convergence to tours. An empirical study of the $A - C$ parameter plane [Hegde *et al.* 1988] also found that the region in which valid tour states were produced decreased as N grew, and even at $N = 8$ was very small. Of course, it is always possible to force the network into tour states, simply by setting $A, C \gg D$, but this means that the tour-length term has negligible impact, which results in randomly chosen tours.

Given the strong evidence that the original HT algorithm was ineffective, several amendments were proposed. [Brandt *et al.* 1988] for example argued that the

three restraint terms should be replaced by two terms: $\sum_i(\sum_a V_{ia} - 1)^2$ and $\sum_a(\sum_i V_{ia} - 1)^2$, and found that this consistently gave valid solutions for up to 32 cities. Even so, it seems unlikely that this completely cures the ills of HT, given that the average 32 city solution was some 20% longer than that found by human eye. Aiyer has given a theoretical analysis of how the network dynamics are influenced by the eigenvalues of T [Aiyer 1991, Aiyer *et al.* 1990]. That analysis gives a prescription for how T can be altered so that the network is confined to a subspace of the hypercube corresponding to valid solutions. Simulations confirmed that valid solutions were always obtained. Again however, the final tours (for $N = 30$) were of poor quality, being roughly comparable to those from the nearest neighbour heuristic.

Further work within the basic HT framework, such as that above, will possibly yield further improvements to the method. A more promising approach however, is described in the next section.

Normalized Networks: the Potts Approach

Clearly the original HT algorithm had difficulty constraining the network into valid solutions. One way forward is therefore to lessen the computational burden being placed on the network. This can be done by constraining each city to be 'on' only once, by enforcing $\sum_a V_{ia} = 1 \, \forall i$, rather than relying on an energy penalty term to try and do this (as in HT). This idea was first utilised in the TSP by [van den Bout & Miller 1988], with equivalent normalization schemes for HT style networks for the GPP analysed in [van den Bout & Miller 1990], [Hérault & Niez 1989], and [Peterson & Söderberg 1989], with the last paper being the only one also analysing the TSP case. For the TSP it greatly enhances the degree of convergence to valid solutions.

In all of these papers the normalization occurs in the context of a 'mean field

annealing' approach. This generally yields the same final network equations as the 'neuronal circuit' approach of [Hopfield 1984], but the exposition is clearer, as it is laid out in a statistical mechanics framework.[2] The idea is to regard each hypercube vertex as a configuration, with an associated E value. The partition function is then formed by summing the Boltzmann factors of the admissible configurations. (For example, in the HT algorithm, all of the 2^{N^2} configurations are admissible, whereas if each city is restricted to being visited only once, then only N^N vertices are admissible.) After taking a mean field approximation, saddlepoint equations are derived, the solutions of which pick out the dominant states of the network at the current temperature T. For the HT model, these saddlepoint equations are equivalent to the network equations 1.8, with E' in (1.9) identified as the free energy and the gain λ acting as an inverse temperature. Analogous correspondences hold for other networks. The mean field method will be covered more fully in chapter 4.

This statistical mechanics characterisation gives a clearer understanding of why the normalized model (which corresponds to a type of Potts model [Wu 1982] in physics, in that each city is represented by a spin which can be in only one of N states) ought to perform better than the HT model. As [Peterson & Söderberg 1989] indicated, the HT algorithm's partition function sums over a vast number of configurations which are nothing like tour states, and even though these offenders have small Boltzmann weights their large number inevitably affects the thermal average quantities. This effect, though not entirely eliminated, is much reduced in the Potts network. A second advantage of this derivation is that, by introducing a temperature, the concept of annealing i.e. reducing T during a run, can be justifiably employed.

Chapters 4 and 5 study the mean field Potts TSP algorithm introduced and first studied by [Peterson & Söderberg 1989]. Dynamical issues, parameter choice and performance are all investigated. We shall find that although the net-

[2]This equivalence was also noted in [Hopfield & Tank 1985].

work's behaviour is quite complex, the key relationships between the various free parameters can be elucidated. However, questions still remain about the algorithm's scalability for very large problems.

Topographic Mapping TSP Algorithms

These methods are derived from algorithms which construct topographic — i.e. neighbourhood preserving — maps between different geometrical spaces, possibly of different dimensionality. As such, they are only applicable to the Euclidean TSP (or other geometrical optimization problems), a drawback partially compensated for by the fact that the progress of the algorithms can be visualised. The methods have a common ancestry in that they are all derived from original research on the theory of topographic mapping. This was concerned with understanding how, in biological systems, connections could develop from a sensory feature 'space' e.g. the retina, onto a cortical (or mid-brain) space (typically of equal or lower dimension than the presynaptic sensory space), so that neighbouring points in the cortex are tuned to points close together in the sensory space.

One such neurobiological model is the 'Tea Trade model' [Willshaw & von der Malsburg 1979], in which a 2-D space maps onto another 2-D space. [Durbin & Willshaw 1987] simplified it, reduced the target dimensionality to 1, and showed how it could be applied to the 2-D TSP. The 1-D structure is a closed loop of points which can be viewed superimposed upon the 2-D city space. The loop points, which ultimately define the tour, move in parallel, under dynamics which gradually pull them towards the cities whilst simultaneously trying to keep the neighbouring loop points close together. This so-called 'elastic net' method found better tours than the original HT algorithm, as well as exhibiting far less sensitivity to parameter choice.

Three further features of the method are its inherent parallelism, the existence of an energy function mimimised by the dynamics, and a complexity of usually $O(N^2)$ per update, compared to $O(N^3)$ for HT. Further detail on the elastic net TSP algorithm and relevant research is given at the start of the next chapter, which studies parameter choice issues in the algorithm. In other related research, [Frean 1990] made an ingenious attempt to adapt the elastic net to deal with non-Euclidean TSPs, with partial success; and [Aue 1990] adapted it to another $\mathcal{NP}$-hard problem, the Euclidean Steiner Problem. Finally, [Goodhill 1992] has taken the method and applied it back into neurobiology, in an attempt to model the joint development of topographic and ocular dominance maps in the cortex.

Another group of mapping algorithms, commonly referred to as 'self organizing feature maps' or 'Kohonen-style' algorithms, share some aspects with the elastic net; see for example, the algorithms of [Angéniol *et al.* 1988] and [Fritzke & Wilke 1991]. Algorithms of this sort are usually derived from the self-organizing mapping specified in [Kohonen 1988], which was itself inspired by the 'Tea Trade model'. They are similar to the elastic net in that they also evolve a loop of points towards the cities but differ in that they operate by selecting single cities at random and then moving the bead nearest to that city towards the city. To keep the loop tight, the winning bead's neighbours also move in sympathy with it; with the width of the neighbourhood decreasing over time somewhat analogously to an annealing temperature. These serial dynamics preclude the existence of any simple energy function for the model, and would also make parallel implementation problematic. This latter point may not be important however, given that the run time complexity is usually low: [Fritzke & Wilke 1991] reports tours of 2-Opt quality, obtained in linear time. Whether such impressive time complexity can also be carried over into algorithms for other combinatorial optimization problems, or indeed whether any such problems can be tackled at all by the self-organizing algorithms, remains

to be seen.

1.6 Theoretical Results on the TSP

A great many mathematical results are known about the distribution of tour-lengths for certain types of TSP — for example, results on the average, worst, and optimal tours, as well as asymptotic N results. Only one will be quoted here however. Most of the experimental testing of new algorithms is carried out first on a class of Euclidean instances in which the N points are independently chosen from the uniformly random distribution on $[0,1]^d$, i.e. the d-dimensional unit cube. Under these conditions [Beardwood *et al.* 1959] proved that, letting $\mathcal{L}_N$ denote the length of the optimal tour through any such set of N points, there is a constant c_d such that with probability 1

$$\lim_{N\to\infty} \mathcal{L}_N = c_d N^{1-1/d} \tag{1.11}$$

Although proving the convergence to a constant is hard (see e.g., [Beardwood *et al.* 1959], or the 'simplified' proof in [Steele 1990]), it is quite easy to understand the scaling relationship, simply by noting that if the points were arranged in a regular lattice, the spacing would be $N^{-1/d}$ and hence $N^{1-1/d}$ would be a lower bound on the optimal tour. Although the exact values of the c_d terms remain unknown, many bounds have been deduced (see e.g., [Steele 1990] and references therein). Numerical studies [Bonomi & Lutton 1984] show that $c_2 \approx .75$. This value will be used to derive estimates for optimal tour-lengths (for unit square problems) whenever necessary in the rest of the thesis.

Statistical Mechanics Studies

This thesis focuses on optimizing the performance of neural networks designed to find low cost solutions to specific TSP instances. Statistical mechanics tech-

niques, on the other hand, allow investigation of the average properties of systems in the $N \rightarrow \infty$ limit. Despite this distinction, results from statistical mechanics are still of considerable interest in interpreting and potentially improving the performance of optimization neural networks. The following remarks serve briefly to cover some of the main results and references from statistical mechanics.

Ideally we would like a complete understanding of the Ising and Potts style spin systems which underly the Hopfield-Tank and Potts optimization networks, with, for example, results on the existence or otherwise of phase transitions, and how the near-optimal solutions are distributed in configuration space. To achieve this however, would require the use of a Hamiltonian comprising several additive terms, one being the solution cost for valid configurations, with the others serving to penalize spin configurations which do not satisfy the problem's constraints. There are considerable difficulties in the analysis of such systems [Fu & Anderson 1986], and to the current author's knowledge such an analysis has not yet been achieved for any network designed for an $\mathcal{NP}$-hard optimization problem. An analytical treatment has proved possible however for a Hopfield-Tank system for the easier 'assignment' problem [Kastella 1992].

More progress can be made if the possible states of the system are restricted to being valid solutions of the optimization problem, so that the Hamiltonian simply gives the solution cost. This case is clearly of direct relevance to simulated annealing algorithms, which can also be used to test the resulting analysis. For geometric unit square TSPs, [Bonomi & Lutton 1984] gave an annealed approximation analysis and demonstrated that it gave an excellent description of experimental data (on the internal energy, specific heat, and entropy as functions of temperature) for a 400 city problem at all but very low temperatures. The breakdown of the approximation in the low temperature regime is unfortunate, as this is the regime of prime interest with respect to near-optimal tours. The transition between the high and low temperature regimes, in which the average

tour-length changes its scaling behaviour with N, appears to occur gradually rather than through a phase transition.

[Kirkpatrick & Toulouse 1985] performed a similar analysis, with similar results, for the case of symmetric non-Euclidean TSPs in which the distances are random variables drawn uniformly from $[0, 1]$. Non-Euclidean problems are theoretically easier to work with than Euclidean ones, as the inter-city distance terms are uncorrelated. Even for this class though, attempts to analyse the low temperature regime using replica methods have met with only limited success [Baskaran *et al.* 1986]. Nevertheless, [Krauth & Mézard 1989] have managed to show — through a non-replica zero temperature analysis — that random $[0, 1]$ problems have optimal length ≈ 2.014 in the thermodynamic limit, a result which we shall use later in chapter 5.

CHAPTER 2

Parameter Sensitivity of the Elastic Net Algorithm

This chapter deals with the elastic net [Durbin & Willshaw 1987], a connectionist algorithm for the Euclidean TSP. This algorithm is capable of finding good solutions, however, for certain ranges of parameter values, it can converge into local energy minima which do not correspond to valid tours. The key parameter is the ratio governing the relative strengths of the two competing terms in the elastic net energy function. Building on recent work [Durbin *et al.* 1989], the parameter regime in which the net may visit some cities twice is first examined. Further analysis predicts the regime in which the net may fail to visit some cities at all. Understanding these limitations allows one to select the parameter value most likely to avoid either type of problem. Simulation data supports the theoretical work.

2.1 Introduction

The elastic net can be visualised as a rule for deforming an imaginary elastic band placed in the city plane by attractive, distance dependent forces from the cities and by elastic forces within the band itself. A scale parameter controls the effective range of the city forces. It is initially set high, then gradually reduced; thus it plays a rôle comparable in some respects to 'temperature' in simulated annealing. In practice, the net is modelled by a finite number of points ('beads') and the algorithm reduces to an iterative procedure for updating the bead positions.

Let $\boldsymbol{x}_i$ denote the fixed position of the ith city, $1 \leq i \leq N$, and $\boldsymbol{y}_j$ the variable position of the jth bead, $1 \leq j \leq M$, $M \geq N$. At each iteration all of the beads are updated in parallel by:

$$\Delta \boldsymbol{y}_j = \alpha \sum_i w_{ij}(\boldsymbol{x}_i - \boldsymbol{y}_j) + K\beta(\boldsymbol{y}_{j+1} - 2\boldsymbol{y}_j + \boldsymbol{y}_{j-1}) \tag{2.1}$$

where α and β are the constants governing the strengths of the city and tension-like forces respectively and K is the scale parameter. w_{ij} — the normalized 'weight' of the connection between the ith city and jth bead — is defined by

$$w_{ij} = \frac{\phi(|\boldsymbol{x}_i - \boldsymbol{y}_j|, K)}{\sum_k \phi(|\boldsymbol{x}_i - \boldsymbol{y}_k|, K)} \tag{2.2}$$

where $\phi(d, K) = e^{-d^2/2K^2}$. This update rule performs gradient descent on an energy function E defined below. Alternative minimization methods and their corresponding update rules are analysed in chapter 3.

$$E = -\alpha K \sum_i \ln \sum_j \phi(|\boldsymbol{x}_i - \boldsymbol{y}_j|, K) + \frac{\beta}{2} \sum_j |\boldsymbol{y}_{j+1} - \boldsymbol{y}_j|^2 \tag{2.3}$$

Note that the second term has coefficient $\beta/2$, and not β, as is stated in the original and several subsequent papers. For net configurations corresponding to tours, the energy reduces to the second term in (2.3) as $K \to 0$. This term is approximately related to the square of the total tour-length, so that in the

limit of $K \to 0$, the deepest minima in the energy landscape tend to represent the shortest tours. The algorithm attempts to find one of these minima by first finding a minimum at high K, where the energy landscape is smoother, then trying to track it as K is reduced.

As [Simic 1990] pointed out, summing over the inter-bead distance-squares rather than over the distances in the second energy term is not ideal. Minimization of the former sum only corresponds to minimization of the tour-length in the limit $M/N \to \infty$, when, assuming the M beads are equally spaced, the sum is proportional to the square of the tour-length. This drawback was alluded to in the original paper but was not stressed therein.

2.2 A Brief Review of Previous Elastic Net Work

[Simic 1990] and [Yuille 1990] independently demonstrated that the elastic net and the method due to [Hopfield & Tank 1985] are related through a common underlying framework, Simic through statistical mechanics and Yuille from work on stereo vision models.

Durbin, Szeliski and Yuille [Durbin *et al.* 1989] (referred to as DSY hereafter) investigated how the energy landscape changes as K is reduced, and deduced several results. Firstly, that for E to remain bounded, every city requires at least one bead within a distance of $O(K^{1/2})$ of it; secondly, a condition on β/α is needed to prevent two neighbouring beads converging on the same city. Thirdly, they derived an implicit expression for the critical value of K, K_c, above which the energy function has a minimum corresponding to all the beads lying at the centre of the city distribution, and they proposed using K_c and this configuration as the initial state. Finally, DSY discussed how the system's dynamics are influenced by the Hessian of E, and concluded that the algorithm

cannot guarantee to find the global minimum, even when the initial state is chosen in this way.

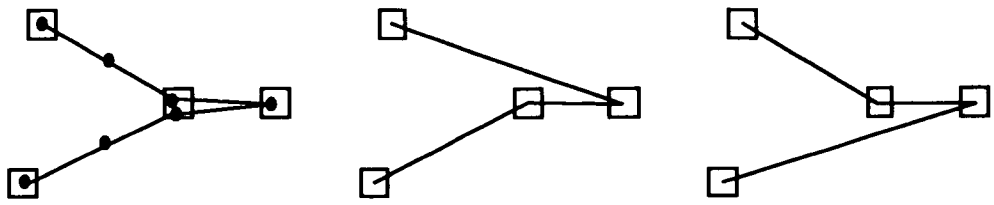
The elastic net algorithm contains few parameters and so it might be possible to understand through analysis what the good parameter settings are, as, for example, DSY did for the initial value of K. One of the most important parameters is the ratio β/α, referred to as γ in this and the following chapter, which controls the relative strength of the tension-like forces to the city forces. This chapter builds upon the work of Durbin *et al.* on selecting the value of γ. I prove that, as $K \to 0$, there exist local energy minima in which some cities remain unvisited, i.e. minima corresponding to configurations that are not tours. The range of γ values for which the algorithm is liable to find one of these minima (and therefore fail) is then derived. This information, combined with the earlier DSY condition, gives a good prescription for choosing γ as a function of the typical separation (denoted by μ) between neighbouring beads in a tour configuration.

2.3 Sensitivity to the Value of β/α

DSY investigated the parameter conditions needed to guarantee that, as $K \to 0$, there would be only a single bead at each city. They analysed the stability of equilibrium configurations of two beads close to a single city; K was considered small enough such that only these beads interact significantly with the city. From their work, the condition for instability (hence for only one bead at the city) can be expressed as

$$\left(\frac{\alpha}{\beta}\right)^2 < \left(\frac{w_2}{w_1^2}|\boldsymbol{A}_1|^2 + \frac{w_1}{w_2^2}|\boldsymbol{A}_2|^2\right) \tag{2.4}$$

where $\boldsymbol{A}_j = (\boldsymbol{y}_{j+1} + \boldsymbol{y}_{j-1} - 2\boldsymbol{y}_j)$ and w_j is the weight between the city and jth bead. They then considered only the case in which the two beads are immediate neighbours in the net. This allows $|\boldsymbol{A}_j|$ to be interpreted approximately as the distance separating a bead converging on the city from its neighbour which is



(a) (b) (c)

Figure 2.1: Example of spiking. Open squares denote cities, dots beads. (a) Net configuration showing a spike caused by two non-neighbouring beads converging onto one city. (b) and (c) show the two possible city orderings obtainable by effectively 'deleting' one of these beads.

---

not attracted to the city. Thus the $|\boldsymbol{A}_j|$ terms can be approximated by μ. Now, the minimum of $(\frac{w_2}{w_1^2} + \frac{w_1}{w_2^2})$, subject to $(w_1 + w_2) = 1$ (which follows from the above assumptions and equation 2.2) is 4; it can be inferred therefore that to prevent two neighbouring beads converging onto a single city, α and β should be chosen such that

$$\frac{\beta}{\alpha} > \frac{1}{2\mu} \tag{2.5}$$

However, preventing the convergence of neighbouring beads onto the same city is not strictly necessary, since such a configuration still defines a perfectly valid tour. Suppose however that the beads are not neighbours: equation 2.4 still holds but the $|\boldsymbol{A}_j|$ terms can now become arbitrarily small, depending on the relative locations of the beads and their neighbours. Hence $\frac{\beta}{\alpha}$ may need to be arbitrarily large to prevent the convergence of both beads and subsequent formation of a 'spike' in the net (see Fig.2.1).

A tour configuration containing a spike is, strictly speaking, an illegal tour, since the city at the spike's base is visited twice, but a simple post-processing operation can recover a legal tour (see Fig.2.1). Indeed, such an operation occurs naturally in the procedure detailed in section 2.5 to extrapolate a city tour from

the final network state.

In summary, satisfying $\gamma > \frac{1}{2\mu}$ should ensure that no city will have two neighbouring beads close to it as $K \to 0$, but it is no guarantee against spikes in the net. All that can be predicted from this analysis about the spike problem, is that their frequency should be a decreasing function of γ. The issue of how to estimate μ for any particular problem will be discussed at the end of section 2.4.1.

2.4 Stable Non-Tour Configurations in the $K \to 0$ Limit

The previous section suggested that to avoid spikes γ should be chosen 'large'. This section will demonstrate that such a policy can cause other problems. To motivate what follows, observe that even in simple situations the algorithm can fail to find a net configuration which visits every city (see Fig.2.2). Such failures can occur in situations where two (or more) cities lie close together. An insight into why this may happen can be gained using the result from DSY that, for E to remain bounded, every city requires at least one bead within a distance $O(K^{1/2})$ of it. During the early stages of the algorithm a close pair of cities may not be resolvable on a length scale of $O(K^{1/2})$. Thus the system may only commit one bead to the region yet still be able to keep the energy contribution of both cities bounded. Later as $K \to 0$ and the cities become resolved, the bead converges to the point midway between the cities. Figure 2.3 is a rough sketch of this.

To prove the stability of the midpoint configuration, consider the situation of Fig.2.3c in detail. Let the cities lie at $(\pm\Delta, 0)$ and consider the component E' of E due to these cities as $K \to 0$. The contribution of the second term in the energy function (2.3) can be ignored since, being only of $O(\beta)$, it will be shown below to be negligible compared to the other term as $K \to 0$. Let the closest bead lie at

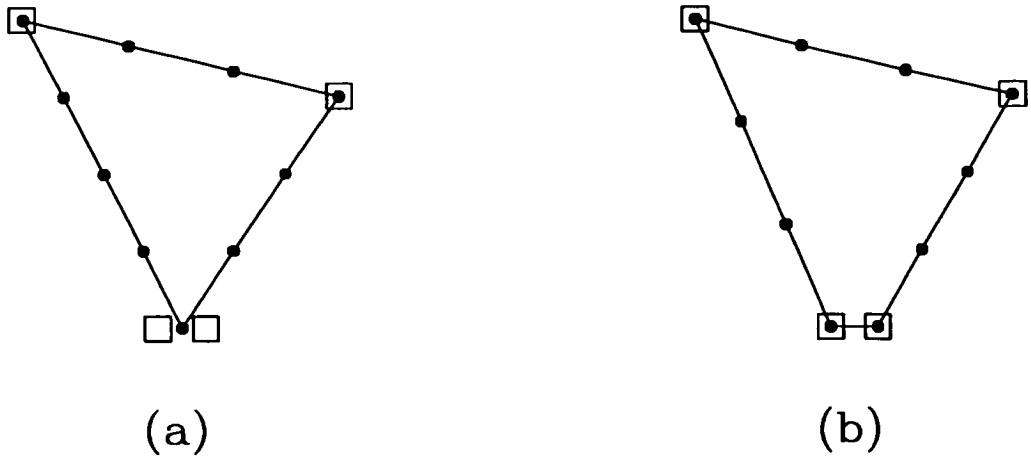


Figure 2.2: Failure of the algorithm on $N = 4$. Squares represent the cities (with coordinates of (.3, .7), (.7, .6),(.47, .3) and (.53, .3)), dots represent the beads. (a) Final configuration found using 10 beads, $\alpha = .05$, $\beta = 1.0$, with an initial state having $K > K_c$ and the beads configured in a small ring around the centre of the cities; K was reduced by 1% every 20 updates. A slower K reduction schedule (1% every 100 updates) also found the same configuration. (b) Final configuration found using the same parameters as (a), except that here $\alpha = 0.1$.

(x, y); the other more distant beads can be ignored since these have negligible weights with the two cities in the $K \to 0$ limit, a fact easily established from (2.2). Thus

$$\begin{aligned} E' &= -\alpha K \left[\ln\left(e^{\frac{(x-\Delta)^2+y^2}{-2K^2}}\right) + \ln\left(e^{\frac{(x+\Delta)^2+y^2}{-2K^2}}\right) \right] \\ &= \frac{\alpha\Delta^2}{K} + \frac{\alpha}{K}(x^2 + y^2) \end{aligned} \tag{2.6}$$

This shows that the bead lies in a radially symmetric quadratic well, the minimum of which is midway between the cities, and that the energy of the (stable) equilibrium configuration rises without bound as $K \to 0$.

This disproves the previous claims of [Durbin *et al.* 1989] that in the limit of small K all minima correspond to valid tours, since it shows the existence of energy minima corresponding to configurations in which some cities remain

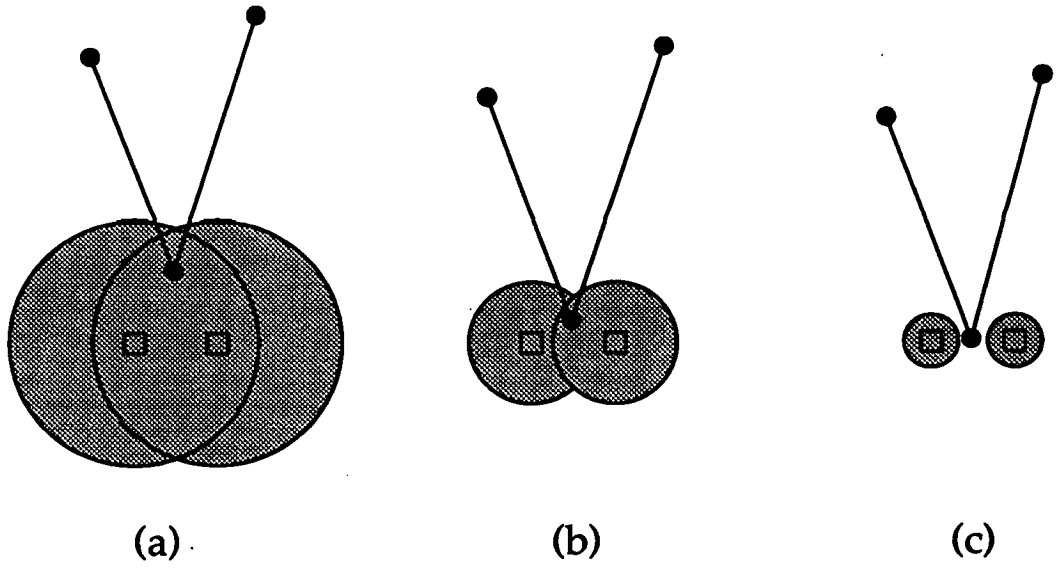


Figure 2.3: Resolution argument to account for failure on a close pairs of cities. Squares denote cities, dots beads and the shaded disks are schematic representations of the $O(K^{1/2})$ zones of each city (see main text for details). $K_1 > K_2 > K_3$. (a) Situation at K_1, cities not resolvable. (b) Situation at K_2. (c) K_3, cities are resolved. The neighbouring beads cannot move in because as $K \to 0$ their weights with the cities become negligible.

---

unvisited (a city is considered 'visited' if, for any small distance ϵ, some bead(s) can be found within ϵ of it in the $K \to 0$ limit).

It might be suggested, notwithstanding the plausibility argument sketched in Figure 2.3, that the trajectory leading to the Figure 2.2a configuration would only be selected if the cooling was too rapid for the system to pick out the 'proper' path leading to the Figure 2.2b configuration. Such an optimistic view would, however, be misguided: as mentioned in the Fig. 2.2 legend, cooling more slowly yields the same results. Furthermore, this reflects the observation that at intermediate K the Fig. 2.2a trajectory has the lower energy. This was seen by repeating the Fig. 2.2 simulations with values of α just above and below the value (≈ 0.825) demarcating the transition between the two types of final state (so that the energy functions are comparable), and examining the two

E vs.K plots. Thus the problem of non-tour minima is fundamentally due to the energy function, and not to any problems in the dynamics or cooling schedule.

2.4.1 Avoiding Non-Tour Minima — β/α Revisited

There is a straightforward way to avoid such minima: simply use so many beads that the typical spacing between neighbouring beads is much less than the minimum inter-city spacing. This strategy is, however, inefficient, since the computational complexity per iteration is $O(NM)$. Instead, the analysis presented below will show that non-tour minima can be avoided by choosing γ such that $\gamma < 1/2\mu$.

Consider the stability of an equilibrium configuration in which two beads (labelled 1 and 2, though this does not imply that they are neighbours) lie near a close pair of cities, for small K. If the configuration remains stable as $K \to 0$ then each city can attract a bead to it; instability, however, leaves just one bead with the cities and subsequently the system becomes trapped in a non-tour energy minimum. Let the equilibrium distances from the beads to the cities be s_1 and s_2 respectively, with corresponding weights w_1 and w_2. The analysis below considers the general case in which these weights are not necessarily equal, though eventually it will become clear that the $w_1 = w_2$ case is the most relevant one. Several simplifying assumptions shall be made here. Firstly, these beads interact significantly only with these two cities. Similarly, these two cities interact significantly only with these beads. Secondly, the two cities are assumed to be coincident; this simplifies the analysis and also represents the 'hardest-case' local scenario for the algorithm in its attempts to have every city visited by a unique bead.

We seek the conditions for which this equilibrium system is a local minimum,

by examining the change in energy induced by local perturbations of the beads. Without loss of generality, let $s_1 + \delta_1$ and $s_2 + \delta_2$ denote the distances between the cities and beads 1 and 2 respectively, in the perturbed state. It will be helpful here to write $E = E_1 + E_2$ with $E_1 = -\alpha K \sum_i \ln \sum_j e^{-|\boldsymbol{x}_i - \boldsymbol{y}_j|^2/2K^2}$ and $E_2 = \beta/2 \sum_j |\boldsymbol{y}_{j+1} - \boldsymbol{y}_j|^2$. To derive ΔE_1, the change in E_1, let $C = \ln(\sum_{k=1}^{2} e^{h(s_k+\delta_k)^2})$, and observe that C can be expressed as

$$\begin{aligned} C &= \ln\left[e^{hs_1^2}e^{h({\delta_1}^2+2s_1\delta_1)} + e^{hs_2^2}e^{h({\delta_2}^2+2s_2\delta_2)}\right] \\ &= hs_1^2 + \ln\left[e^{h({\delta_1}^2+2s_1\delta_1)} + \Omega e^{h({\delta_2}^2+2s_2\delta_2)}\right] \\ &= hs_1^2 + \ln(1+\Omega) + \ln(1+\psi), \end{aligned} \tag{2.7}$$

where

$$\psi \equiv \frac{2h(s_1\delta_1 + \Omega s_2\delta_2) + h(\delta_1^2 + \Omega\delta_2^2) + 2h^2(s_1^2\delta_1^2 + \Omega s_2^2\delta_2^2) + O(\delta_*^3)}{1+\Omega}, \tag{2.8}$$

with δ_* and Ω denoting $\max(|\delta_1|, |\delta_2|)$ and $e^{h(s_2^2 - s_1^2)} (\equiv w_2/w_1)$ respectively. Expanding $\ln(1+\psi)$ to second order in δ_* and gathering like terms gives

$$\begin{aligned} C &= hs_1^2 + \ln(1+\Omega) \\ &\quad + \frac{1}{1+\Omega}\left\{\frac{2h^2\Omega}{1+\Omega}(s_1\delta_1 - s_2\delta_2)^2 + h(\delta_1^2 + \Omega\delta_2^2) + 2h(s_1\delta_1 + \Omega s_2\delta_2)\right\} + O(\delta_*^3) \end{aligned} \tag{2.9}$$

Setting $h = -1/2K^2$ and noting that at equilibrium (2.1) implies $s_k = \gamma A_k K/2w_k$ (where $A_k = |\boldsymbol{A}_k|$), therefore gives ΔE_1 correct to second order as

$$\Delta E_1 = \frac{\alpha\gamma\delta}{1+\Omega}\left(\frac{A_1}{w_1} - \frac{\Omega r A_2}{w_2}\right) - \frac{\alpha\delta^2}{K(1+\Omega)}\left[\frac{\Omega\gamma^2}{4(1+\Omega)}\left(\frac{A_1}{w_1} + \frac{rA_2}{w_2}\right)^2 - (1+\Omega r^2)\right] \tag{2.10}$$

where the perturbation has been parameterised by writing $\delta_1 = \delta$, $\delta_2 = -r\delta$. Noting that, by definition, the first order component of ΔE vanishes at an equilibrium state, and, by inspection of the E_2 definition, that the second order component of ΔE_2 will contain no K dependence, we find

$$\Delta E \approx -\frac{\alpha\delta^2}{K(1+\Omega)}\left[\frac{\Omega\gamma^2}{4(1+\Omega)}\left(\frac{A_1}{w_1} + \frac{rA_2}{w_2}\right)^2 - (1+\Omega r^2)\right] \tag{2.11}$$

as $K \to 0$. Thus ΔE is positive provided γ satisfies

$$\gamma^2 < \frac{4\Omega(1+r^2\Omega)}{(1+\Omega)(\Omega A_1 + rA_2)^2}. \tag{2.12}$$

Differentiating with respect to r, the r.h.s. is found to have a single minimum at $r = A_2 A_1^{-1}\Omega^{-2}$. Evaluating the r.h.s. at this point gives the condition for stability against the whole range of perturbations as

$$\gamma^2 < \frac{4\Omega^2}{(1+\Omega)(\Omega^3 A_1^2 + A_2^2)}. \tag{2.13}$$

Switching from the Ω notation back into that of the weights, noting that $w_1 + w_2 = 1$ and writing w_1 as w, we can therefore guarantee that ΔE is always positive, and hence that the configuration is stable, in the low K limit, by choosing γ to satisfy

$$\gamma A' < \sqrt{\frac{4w^2(1-w)^2}{3w^2 - 3w + 1}}; \quad A' = \max(|\boldsymbol{A}_1|, |\boldsymbol{A}_2|) \tag{2.14}$$

The right hand term of this inequality is a single-humped function, symmetrical about $w = 1/2$. As $w \to 0$ or 1 this function goes to zero, implying that γ may unfortunately need to be chosen arbitrarily small to prevent instability. This is just a formal expression of the idea in Figure 2.3, that, once a single bead begins to dominate the interaction with the pair of cities this dominance tends to grow, so that as $K \to 0$ this bead is the only one close to the cities. The crucial point therefore is to prevent the emergence of a single dominant bead in the first place, by ensuring that configurations having two beads with comparable weights remain stable down to the $K \to 0$ limit.[1] Thus the case of $s_1 = s_2$ (i.e. $w = 1/2$) is the most relevant one for getting a constraint on γ. In this case (2.14) gives the stability condition as $\gamma < 1/A'$. When the two beads are immediate neighbours in the net $|\boldsymbol{A}_k|$ is approximately μ, whereas for cases in which the beads are not neighbours $|\boldsymbol{A}_k|$ can clearly range from approximately 2μ down to zero, where μ is as defined in section 2.2. Thus, the prediction of this analysis is that all non-tour minima can be avoided by selecting γ such that $\gamma < 1/2\mu$.

[1] Or, in the case of a close but non-coincident pair of cities, stable down to the K value at which each bead converges to a specific city.

Two remarks should be made about the above energy analysis and its result. Firstly, equation 2.14 can also be derived by modifying DSY's eigenvalue analysis of the two beads/one city configuration to the current two beads/two cities case. Similarly, an energy analysis of the two beads/one city case yields the instability condition found by the earlier DSY eigenvalue analysis. This correspondence arises since, whereas the energy analysis determines directly whether the equilibrium state is a local minimum, DSY's analysis does this indirectly, essentially by investigating the eigenvalues of the Hessian. Secondly, the fact that $1/2\mu$ is an upper bound on γ for the stability of the two beads/two cities case as well as the lower bound for the instability of the one city/two (neighbouring) beads case is largely explicable in that these two cases are clearly mathematically related.

In summary, $\frac{1}{2\mu}$ emerges as an important value for the parameter γ (or β/α). Choosing γ below $\frac{1}{2\mu}$ risks creating spikes in the net as well as the lesser problem of neighbouring beads converging on the same city; whilst setting γ above $\frac{1}{2\mu}$, though it decreases the likelihood of spikes, risks the system finding a non-tour minimum.

Since μ is the average separation between neighbouring beads, it can be estimated given some prior estimate of the tour-length. For instance one can use the result of [Beardwood *et al.* 1959], discussed in section 1.6 — that for N cities drawn randomly from the unit square, the optimal tour has length $c\sqrt{N}$ in the $N \to \infty$ limit, with $c \approx .75$ — to give a crude tour-length estimate for TSPs of that class, even for non-asymptotic N. For TSPs in which the spatial distribution of cities is not homogeneous, μ will also likely show spatial variation. So in this case, assuming some tour-length estimate, to achieve a consistently optimal value of γ, the value of β would need to vary between different parts of the net, a possibility originally noted by Durbin and Willshaw though in a looser context than here. However, TSPs with cities drawn from inhomogeneous distributions have not been investigated experimentally in the current study.

2.5 Simulations and Discussion

Simulations were performed to test whether the algorithm's behaviour varied with γ in the predicted manner. The simulations were carried out on an AMT DAP, a SIMD style array processor; a brief discussion of how the algorithm was mapped onto this parallel computer is given in Appendix A. Ten TSPs with $N = 50$ and ten with $N = 200$ were studied; with all the cities drawn randomly from within the unit square. Every city set was run with a range of γ values, and the number of spike defects and 'frozen bead' defects (i.e. single beads trapped in high energy, non-tour minima) present at the end of each run recorded; the results are presented in Figure 2.4. The values of γ were chosen relative to γ', where γ' denotes the value of $1/2\mu$ using the Beardwood tour-length estimate plus the further assumption that μ is invariant over the net (i.e. $\gamma' = 2MN^{-1/2}/3$). Based on the analysis of DSY, the initial value of K was chosen to be K_c, where K_c is the positive root of

$$4K^3\gamma \sin^2 \tfrac{\pi}{M} + K^2 N/M - \lambda/M = 0 \qquad (2.15)$$

and λ is the principal eigenvalue of the city distribution's matrix of second order moments. Note that this differs from the original K_c prescription of DSY due to several algebraic errors in that reference. K was reduced by 1% every ten updates. Further technical details, including formal definitions of spike and frozen bead defects, are given in the legend to Figure 2.4.

The plots in Figure 2.4 give consistent support to the analytical predictions. Spike and frozen bead defects dominate the low γ and high γ regimes respectively, with $1/2\mu$ marking the approximate boundary between the two regimes; note that, as expected, some spikes still occur above $1/2\mu$ (Fig.2.4d). Qualitatively, the division into two regimes can be understood from the rôles of α and β as the coefficients of competing terms in the energy function. A low value of β/α emphasises moving the beads closer to the cities rather than minimizing

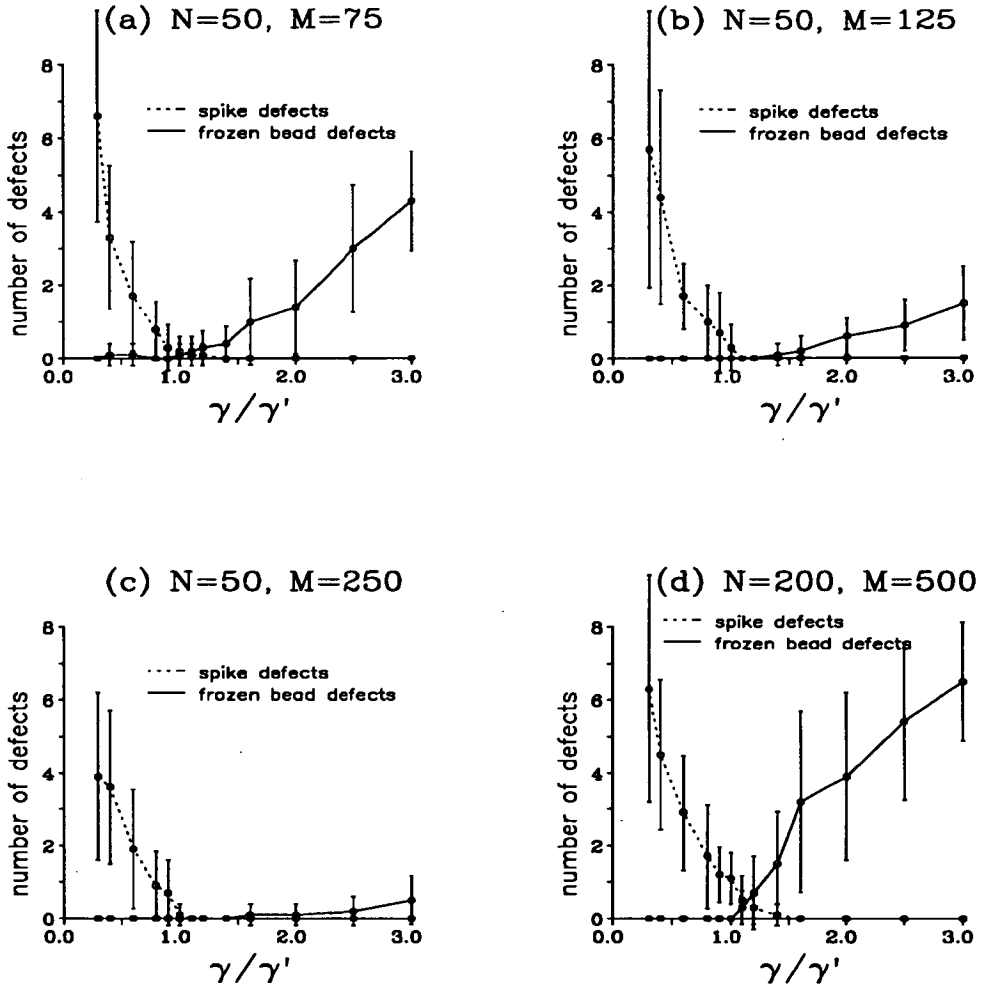


Figure 2.4: Frequency of tour defects as a function of γ/γ'. Each data point represents the mean number of defects (of one type) found in a final net configuration, averaging over ten TSP instances, with the associated standard deviation as the error bar. In all the simulations, β was fixed at 1.0 and the beads were initially placed in a ring of radius 0.05 around the centre of the cities (starting with all the beads exactly at the centre causes problems, because when K is slightly below K_c the gradients there are very small and so the system requires a large number of iterations before settling into an energy minimum). Simulations were terminated when either of two criteria were satisfied: (i) if, $\forall i$, $\max_j(w_{ij}) > 0.95$; followed by a further reduction of K by a decade to allow final settling, or (ii) when $K < 0.01\mu$, μ calculated using the Beardwood estimate discussed in the main text. A **spike** occurs where a city has significant interactions (here taken to mean $w_{ij} > 0.3$) with two or three, non-contiguous beads. A bead k is **frozen** if it is the bead nearest to two or more cities, i.e. if there are two or more cities i for which $\max_j(w_{ij}) = w_{ik}$.

the net length, hence it may lead to cities being visited by more than one bead; a high value of β/α does the opposite, so may lead to some cities remaining unvisited. Figures 2.4(a-c) show that increasing the number of beads M, for fixed N, substantially reduces the number of frozen beads. This trend is understandable since, as mentioned in section 2.4.1, what influences whether a close pair of cities develops a frozen bead defect is not the inter-city distance itself but rather this distance relative to the typical spacing between beads. Increasing M/N appears to have little effect on spiking, except at very low γ where it helps slightly. Finally, the size of the error bars in this figure reflects the fact that although the variation in defect frequency with γ is fairly smooth for any particular TSP, different city sets have differing absolute numbers of defects.

The analyses given here and in DSY plus the discussions of Fig.2.2, strongly suggest that many defects develop because of the intrinsic structure of the energy landscape, and therefore will not just disappear by annealing more slowly. This was confirmed by runs reducing K ten times more slowly than in the Figure 2.4 simulations showing no significant change in the number of defects produced (data not shown). Of course if K is reduced so rapidly that the network has insufficient time ever to relax into local minima (the physical analogy here is of a system cooled too rapidly to allow equilibration at any temperature) then naturally many more defects develop, including frozen beads for $\gamma < 1/2\mu$. This is a point demonstrated in the following chapter.

In summary, to avoid defects γ is best chosen to be approximately $1/2\mu$, or perhaps slightly above this if the ratio of beads to cities M/N is large. If legal tours can be successfully recovered from net configurations with defects using post-processing, then other properties, most obviously the tour-length, may conceivably be optimized by some other choice of γ. No analytical work on this issue was conducted in the current study however. Nevertheless, intuition strongly suggests that nets with many defects (of either variety) will — after post-procesing — give longer tours than those with no, or few, defects, because

in the former case the network has solved a problem different from the TSP. In other words, defects indicate a mismatch between the energy function being minimized and the target problem we wished to solve, making high quality tours unlikely. This contention is supported by Figure 2.5, which shows a measure of relative tour quality against γ/γ', using the following post-processing procedure.

This robust scheme extrapolates from a final elastic net state to a tour. First, construct the set of beads $\{k_i\}_i$, where bead k_i is picked out by the rule $w_{ik_i} = \max_j(w_{ij})$. If this set is non-degenerate i.e. there are no frozen beads, then the reduced net comprising the beads in this set can be trivially extrapolated to a unique tour of the cities.[2] If there is a frozen bead however, then we must somehow construct an ordering around the cities competing for that bead. A simple, random ordering scheme was used in this study.

Figure 2.5 demonstrates empirically that γ' also locates the optimal region of γ with respect to tour-length; the regions either side of γ' giving longer tours, particularly below γ' where spike defects exist. Thus, even if we permit a post-processing stage to remove defects from the net, the analysis begun by DSY and completed herein to determine the optimal parameter setting with respect to avoiding defects is still of value, as the γ value it prescribes also gives short tours.

This work also ties in with Simic's observation [Simic 1990] that the elastic net only solves the 'correct' problem when $M \gg N$. We see here that the consequences of not having $M \gg N$ can include not just sub-optimal tours, but also the possibility (in a particular region of parameter space) of finding net configurations which do not correspond to valid tours at all. The elastic net algorithm is not unique in regard to the possibility of convergence to non-tour configurations — this also happens in the original Hopfield-Tank algorithm (as discussed

[2]Notice that this scheme automatically deals with spike defects.

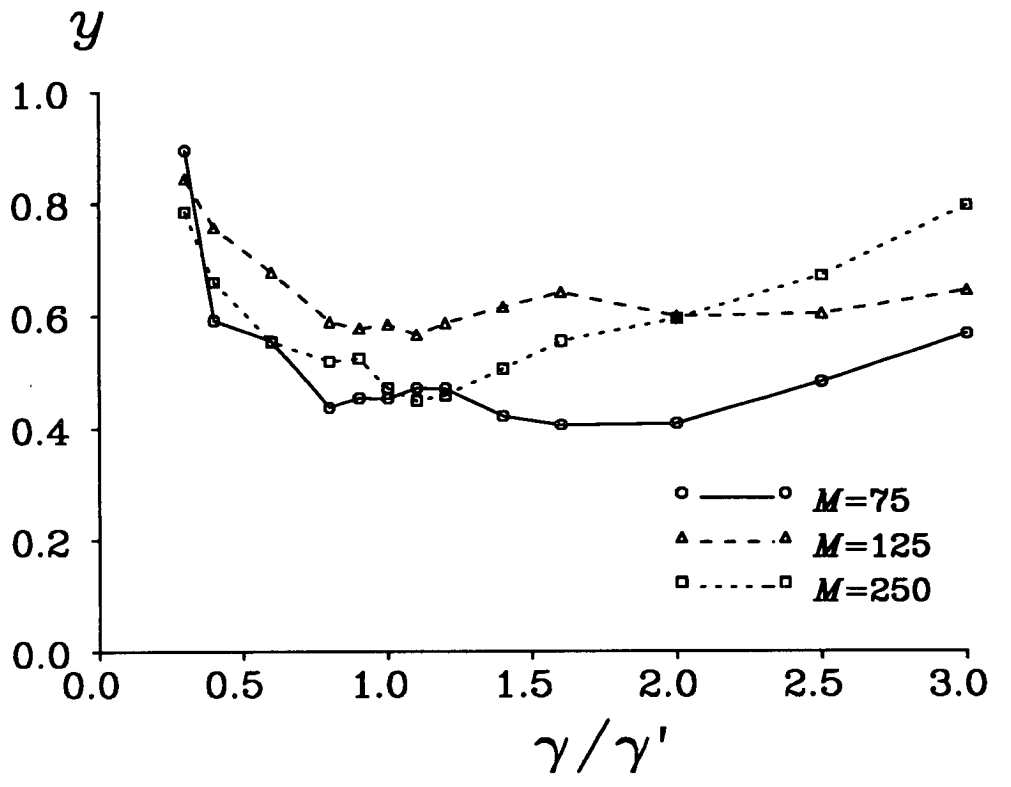


Figure 2.5: Post-processed relative tour quality as a function of γ/γ'. These curves were derived from the Figure 2.4 fifty cities simulation data, using the post-processing scheme detailed in the main text. Letting $\mathcal{E}_\gamma^s$ denote the fraction by which the length of the tour obtained for city set s at parameter ratio γ exceeds the length of the best-known tour, the ordinate y_γ represents $\left\langle \frac{\mathcal{E}_\gamma^s}{\max_\gamma(\mathcal{E}_\gamma^s)} \right\rangle$ where the averaging is over the instances indexed by s. Thus, a value of y_γ close to unity indicates that this value of γ gives tours which are as long as the longest found at any other values of γ. Lower values of y_γ indicate reductions in the average tour-length excess over optimal, relative to that found at the worst value(s) of γ. The normalization step is appropriate because there are sample-to-sample variations in the size of the $\mathcal{E}$ measures. The standard deviations associated with the y_γ terms are still substantial but have been omitted for clarity. The 'best-known' tour is that found over fifty independent trials using the Or*-opt serial algorithm detailed in section 1.4.3. In one case the elastic net tour was shorter than the best Or*-opt tour, and so the former was used as the best-known tour.

in section 1.5) as well as in Peterson and Söderberg's improved Potts version of it, studied in chapters 4 and 5. It should be noted though that the problem is generally less acute in the elastic net and Peterson and Söderberg algorithms, than in Hopfield and Tank's. This is because the Hopfield-Tank energy function has fewer constraints built into it than the energy functions used by the other two algorithms [Peterson & Söderberg 1989, Simic 1990, Yuille 1990]. Although there are reformulations of Hopfield-Tank that do generally converge to tour states (see section 1.5), the resulting tours are of barely acceptable quality even for low N, whereas, as we shall see in the next chapter, the elastic net can generate acceptable tours even for $N = 200$.

2.6 Conclusions

Three particular issues regarding the performance of the elastic net algorithm on the TSP have been addressed here. Firstly, by extending the analysis of [Durbin *et al.* 1989], the problem of cities being visited twice by non-neighbouring beads was examined. Secondly, it was proved that, in the $K \to 0$ limit, there exist high energy local minima in which some cities remain unvisited by the net. Thirdly, the parameter regime in which the algorithm might find one of these non-tour minima was derived. This allowed a decent prescription to be given for the value of the β/α parameter ratio most likely to produce valid tours. For a testbed set of uniformly random square TSPs, simulations were found to support the details of the analysis in all of these areas. Finally, it was argued that even when post-processing to remove defects from the final elastic net state is allowed, the parameter regime which minimizes the number of defects ought also to yield the shortest tours. This idea was also supported by simulation data.

The next chapter investigates other issues regarding the elastic net algorithm, in

particular the use of more sophisticated minimization methods and how these, plus the rate at which K is reduced, affect solution quality.

CHAPTER 3

Enhanced Minimization Methods for the Elastic Net

The original elastic net paper employed the steepest descent method to find minima of the energy function. One could however use one of the many other techniques developed for seeking minima in functions of many variables. Therefore, in assesssing the performance of the elastic net method, it is appropriate to consider whether alternative minimization engines might yield better performance. This chapter examines the performance of the elastic net method when coupled with a variety of minimization schemes which, in keeping with the spirit of connectionism, are *self-adaptive* and largely *local* in character. We shall see that by incorporating knowledge about the energy function into the adaptation process these methods can be made robust, as well as giving better performance than steepest descent.

3.1 Introduction

In deciding which minimization methods to investigate, we shall draw inspiration from similar research in the field of learning algorithms for multi-layer perceptrons, the back-propagation algorithm in particular. The use of locally adaptive step-size methods, for example, is quite common in current back-propagation implementations, and it is perhaps surprising that these (or other improved methods) have not so far been used by researchers developing neural network algorithms for combinatorial optimization problems.

Most of the simulations in this chapter were carried out on TSPs with cities located at points in a square lattice. This was done for two reasons: these grid TSPs have known optimal solutions, and secondly, because they do not have any close pairs of cities, the algorithm will not be prone to the defects studied in the last chapter. Thus these grid TSPs provide an ideal laboratory for studying the tour-length performance aspects of the algorithm. Section 3.2 discusses two simple improvements over steepest descent, namely, the energy monitoring and momentum methods. Section 3.3 studies locally adaptive step-size methods on grid TSPs, whilst section 3.4 looks at their performance on random TSPs of the sort used in section 2.5, the difficulties encountered and one possible strategy for ameliorating them. The remainder of this section explains the drawbacks of the original steepest descent method.

3.1.1 The Original Dynamics

The original update rule (2.1) is equivalent to $\Delta \boldsymbol{y}_j = -K \frac{\partial E}{\partial \boldsymbol{y}_j}$ for the energy function defined in (2.3). A coefficient of $O(K)$ is required to keep the dynamics controlled in the $K \to 0$ limit. As mentioned in the previous chapter, the elastic net method seeks an energy minimum at high K, then tries to track it (or

one of its offspring) as K is reduced. However, it is far from certain that the original implementation, which allows a fixed number of iterations per K value, will realize this aim. The problem is that no attempt is made to ascertain whether the system has actually reached a local minimum before reducing K. In certain types of landscape, steepest descent can have a very slow rate of convergence (see, for example [Jacobs 1988]), so convergence problems should not be unexpected. Problems can occur if K is reduced before the system has relaxed into a minimum. For example, the change in K may lead to the minimum bifurcating; if the system lies on the 'wrong' side of the bifurcation, it will then have no opportunity of entering the deeper minimum on the other side (as the algorithm is deterministic). Drawing on the similarity with simulated annealing, this issue is like that of how to design an annealing schedule that ensures thermal equilibrium is approximately maintained throughout cooling. Studies have shown that schedules which properly monitor the degree to which the generated distribution matches the Gibbs distribution at each temperature are more successful than naïve schedules which simply spend a fixed amount of time at each temperature [van Laarhoven & Aarts 1987]. This experience ought to be translated into the elastic net method, in the context of finding mimima.

One broad strategy for improving the algorithm is to allow a dynamically determined number of iterations per value of K, reducing K only when some convergence criterion is satisfied. Another is to retain a fixed number of iterations per K, but somehow alter the dynamics within this constraint in order to maximize the convergence. We shall be concerned only with methods in the latter category, one reason being that methods using the former strategy have differing run times, which complicates performance comparisons.

The previous chapter discussed the ratio β/α, but did not cover setting appropriate absolute values for α and β. As these values set the scale of E, variations in these will alter the rate of convergence. Alternatively, steepest descent dy-

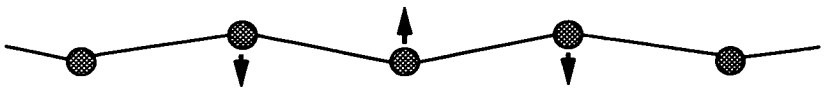


Figure 3.1: Sketch of a net fragment displaced from equilibrium. Filled circles denote beads, with lines connecting neighbouring beads. The arrows indicate the direction of the tension force acting on each bead. If the step-size exceeds a certain value then the oscillation grows, as described in the main text.

namics of the form $\Delta \boldsymbol{y}_j = -\eta K \frac{\partial E}{\partial \boldsymbol{y}_j}$, where η shall be called the step-size, have the same effect, and this is the framework in which the dynamics will be discussed in this chapter. Generally speaking, we want η to be big enough to enable the system to take large steps downhill on the energy surface, but small enough to ensure that the changes in $\boldsymbol{y}_j$ are below the scale of significant features on the surface. An upper bound on the feasible values of η can be derived by considering the situation at an early stage of the procedure, below K_c but before any detailed structure develops in the net.[1] At this stage the city forces are still fairly diffuse, so the dynamics of a small fragment of the net are determined primarily by tension forces, since beads within a small fragment are subject to nearly identical sets of city forces. Figure 3.1 sketches such a fragment, in which the beads are shown slightly displaced from their equilibrium positions. Using the η steepest descent dynamics (and ignoring both the city forces and end-effects in the fragment) it is relatively simple to deduce that if $2\eta\beta K > 1$, successive iterations will cause the kinks to oscillate and grow in amplitude, i.e. the net will become unstable. Simulations undertaken in the current study bear testimony to the importance of this bound, and show that this instability almost always destroys the net, in the sense of bead positions changing between successive iterations on the scale of the city distribution's diameter.

[1]Thanks are due to Marcus Frean for discussions on this point.

The above result places a tight restriction on the values of η that are feasible if η must remain constant throughout a run. However, as the condition $2\eta\beta K < 1$ was derived assuming large K, there is no apparent reason why larger values of η violating this condition cannot be used at lower K. Thus it is natural to consider dynamics in which η is varied in some fashion as K is reduced. The key issue is then deciding what rule(s) to use in varying η, and this constitutes the central theme of the following two sections.

3.2 Energy and Momentum Methods

Rather than keeping η constant or forcing it to vary as a pre-determined function of other parameters (whether derived theoretically or on an ad-hoc basis), the *energy monitoring* method allows it to vary dynamically, in response to the change in energy between successive iterations. The idea here is that if a steepest descent step lowers E, on the next step we use a slightly larger η to try to accelerate the progress towards the minimum. Of course, if unchecked, this mechanism will lead to overshooting of the minimum. So if a step is found to have increased E, that step is rejected, η decreased, and a new step attempted (using the gradients calculated after the last accepted step). Computation of ΔE does not impose a significant overhead because most of the calculation overlaps that of the gradient terms. A multiplicative scheme is used to adapt η i.e. $\eta \mapsto \eta_+\eta$ or $\eta \mapsto \eta_-\eta$, for negative and positive ΔE respectively (the choice of a multiplicative rather than linear rule being motivated by the trio of papers cited in section 3.3). The η_+ and η_- constants are chosen so that $\eta_+ > 1$, $\eta_- < 1$, with $1/\eta_- > \eta_+$ in order to facilitate rapid η reduction when necessary. Values of $\eta_+ = 1.2$ and 1.5 were studied; η_- was fixed at 0.5. To test this method, simulations were performed on lattice TSPs and the degree of tour-length sub-optimality recorded; the data are shown in Table 3.1. Before discussing the data, some remarks about lattice TSPs and terminology are apposite.

	basic steepest descent		*energy monitoring steepest descent*			
			$\eta_+ = 1.2$		$\eta_+ = 1.5$	
N	$K_\tau = .95$	$K_\tau = .99$	$K_\tau = .95$	$K_\tau = .99$	$K_\tau = .95$	$K_\tau = .99$
64	5.2	0.0	5.2	0.0	0.0	0.0
121	6.1	5.4	6.1	3.4	4.8	2.0
196	4.2	4.2	3.4	1.7	1.3	0.8
289	7.7	5.2	5.7	2.6	3.1	1.7
400	6.6	4.1	4.1	0.8	2.5	2.7
Mean	6.0%	3.8%	4.9%	1.7%	2.3%	1.4%

Table 3.1: Tour-length performance data for energy monitoring and original steepest descent methods. The data items are the percentages by which the final tour-lengths exceed the optimal lengths. Technical details: $M = 2.5N$, $\beta/\alpha = 1/2\mu (= MN^{-1/2}/2)$, $\beta = 2.0$, initial state having the beads in a ring of radius .1 and $K = .2$, ten steps per epoch, termination criteria as in section 2.5.

Lattice TSPs have cities at every node of a $p \times p$ square lattice of spacing $1/p$, where $p^2 = N$. Basic geometry shows the tour-length to be bounded below by $\sqrt{N}$ or $\sqrt{N} + (\sqrt{2} - 1)/\sqrt{N}$ for p even or odd respectively, and it is a simple matter to construct tours for any p which achieve the appropriate bound. Henceforth, the time spent at a single value of K will be termed an *epoch*, and the original $\eta = 1$ basic steepest descent dynamics employing a fixed number of updates per epoch as BSD. For the energy monitoring scheme each epoch comprises a fixed number of attempted updates. At the end of each epoch, K is reduced: $K \mapsto K_\tau K$, where K_τ is a constant (< 1) controlling the cooling rate. As the energy surface is deformed each time K is lowered, it is clearly unwise to assume that the value of η at the end of one epoch will be appropriate at the beginning of the next. Therefore η was reset to unity at the start of each epoch for the experiments summarized in Table 3.1.

Table 3.1 shows that the new scheme, compared to BSD, gives substantial reduc-

tions in the average percentage by which the tour-lengths exceed the optimal values. This measure of tour sub-optimality will be used throughout this thesis, being called the 'tour percentage excess', or simply 'percentage excess'. There are, however, at least two objections to the use of the energy monitoring scheme. Firstly, explicit evaluation of E is a global computation, which therefore goes against the grain of the connectionist paradigm. Whilst not adopting a fundamentalist stance on the necessity of purely distributed processing,[2] this is still an undesirable feature, in that one could then argue that other minimization techniques using global computations should also be considered. Our more modest goal is simply to consider how far performance can be enhanced using only local techniques. The second objection is that adapting η on the sign of ΔE alone is quite crude: an overall decrease in energy might mask several localized regions of the net which are developing poorly. A more locally sensitive adaptation technique is required. We shall see in the following sections that local methods can give comparable, and indeed better, performance than the energy monitoring scheme.

The Momentum Strategy

The momentum technique was originally proposed in the context of the back-propagation learning algorithm for multi-layer perceptrons, by [Rumelhart *et al.* 1986]. However, the technique is generally applicable to any minimization problem on which steepest descent can be deployed. For the elastic net algorithm, it corresponds to an update equation of the form

$$(\Delta \boldsymbol{y}_j)^{(n)} = -\eta K \frac{\partial E}{\partial \boldsymbol{y}_j} + \nu (\Delta \boldsymbol{y}_j)^{(n-1)} \quad \forall j \tag{3.1}$$

[2]In its usage here, the elastic net method is not modelling any biological or cognitive process, therefore such a stance is not warranted. Furthermore, it is arguable that the weight normalization in (2.2) already involves an element of global computation.

	momentum value				
N	$\nu = 0$	$\nu = .3$	$\nu = .6$	$\nu = .9$	$\nu = .99$
64	5.2	5.2	5.2	5.2	5.2
121	6.1	6.1	6.1	6.1	6.1
196	4.2	4.2	2.5	4.2	4.2
289	7.7	5.7	5.7	5.7	5.2
400	6.6	4.1	4.1	4.1	5.6

Table 3.2: Tour-length performance data for the momentum method. The data items are the percentages by which the final tour-lengths exceed the optimal lengths. K_r was set at .95; the other parameters were as used for the BSD runs in Table 3.1. At the start of each epoch the $(\Delta \boldsymbol{y}_j)^{n-1}$ terms are regarded as zero.

---

where η is fixed, n labels the update number, and ν — the momentum parameter — is a constant in the range $[0, 1)$.

Adding in a component of the previous update helps convergence in two situations: it dampens out oscillations which steepest descent often generates when narrow 'ravines' are encountered in the energy surface, and it allows the effective step-size to grow for dimensions in which the partial derivative sign remains constant over a wide range. A more detailed discussion of the momentum technique can be found in [Jacobs 1988].

Table 3.2 displays the performance using momentum. The $\nu = 0$ case recovers BSD. The momentum term has negligible impact, except for cases in which both N and ν are high, when it has a slightly beneficial effect. One possible explanation of this result might be that the types of landscapes for which momentum helps may not be present in these grid TSPs. Alternatively, it may be that the particular (η, ν) values studied were far from the optimal one(s) (with regard to convergence). In back-propagation, it was empirically demonstrated

by [Tollenaere 1990] that for any fixed value of ν there is a small range of step-sizes which give rapid convergence. The location of this range varies with ν, the problem type, and the problem size. In the absence of any analytical guidance in setting (η, ν), researchers have developed methods which attempt to find suitable parameters self-adaptively, rather than via brute force parameter space searching. The techniques in the next section were largely inspired by this work in back-propagation.

3.3 Local Step-Size Adaptation Methods

The fundamental idea of these methods is that every dimension of the space on which the energy function is defined should have its own individual variable step-size (or learning rate, in the back-propagation terminology), which adapts in some fashion according to whether the associated partial derivative changes sign between successive updates. Several such methods were discussed by [Jacobs 1988], leading to the refined heuristics of [Tollenaere 1990, 1991] and [Silva & Almeida 1990]. The techniques developed here are closest to Tollenaere's *SuperSAB* algorithm. *SuperSAB* combines the momentum strategy, local step-sizes which adapt exponentially according to the sign of the product between the current and previous partial derivative, and a step rejection mechanism. Only the second of these three elements was utilised in the techniques developed here. In the first such locally adaptive technique studied (termed 'fast adaptation', or FA), the beads were updated in parallel according to:

$$\Delta l_j^{(n)} = -K\eta_{jl}^{(n)} \left(\frac{\partial E}{\partial l_j}\right)^{(n)} \quad \forall j, l \tag{3.2}$$

where, for the 2D geometric TSPs considered here, $l \in \{u, v\}$, with $\boldsymbol{y}_j \equiv (u_j, v_j)$ the position of the j th bead. The superscript n labels the update number. The

step-sizes adapt in the following manner:

$$\eta_{jl}^{(n)} = \begin{cases} \eta_+ \eta_{jl}^{(n-1)} & \text{if } \left(\frac{\partial E}{\partial l_j}\right)^{(n)} \times \left(\frac{\partial E}{\partial l_j}\right)^{(n-1)} \geq 0 \\ \eta_- \eta_{jl}^{(n-1)} & \text{if } \left(\frac{\partial E}{\partial l_j}\right)^{(n)} \times \left(\frac{\partial E}{\partial l_j}\right)^{(n-1)} < 0 \end{cases} \quad \forall j, l \qquad (3.3)$$

As in section 3.2, the η_+ and η_- terms are constants > 1 and < 1 respectively, though of course now they are applied on the basis of purely local partial derivative information. A change in the partial derivative sign indicates that the system is in the vicinity of a local minimum for that dimension, and the above rule responds to this by proceeding more 'cautiously' in that dimension, i.e. by reducing the step-size. Following the reasoning in section 3.2, the $\{\eta_{jl}\}$ are reset to unity at the start of every epoch.

Data on the performance of the FA method on the usual lattice TSPs are presented in the left hand column of Table 3.3. Comparing its average performance with that obtainable using BSD (see Table 3.1), not only does FA produce shorter tours than BSD at the same K_r, it also outperforms BSD runs employing a much slower cooling schedule. Furthermore, the values of the $\eta_{+/-}$ adaptation rates used in these runs were not arrived at by extensive parameter space tuning: further exploration would likely yield even better results. Thus the FA strategy is clearly of merit.

How can the FA strategy be improved upon? There are two related criticisms of the method that point to the answer.

Firstly, the $\{\eta_{jl}\}$ are currently all reset to one at the start of each epoch. However, this reset value was chosen only so as to bring FA into line with BSD at the start of each epoch; other choices are possible. As it sets the absolute scale for the step-sizes, the reset value does need to be chosen carefully. Although it is no longer strictly valid, the $\eta\beta K < 0.5$ condition derived in section 3.1.1 can be used for guidance here. Alternatively, one can select a reset value higher than suggested by the $\eta\beta K$ condition and rely instead on the adaptation mechanism

N	FA	SA	$TLA\ (\equiv FA + SA)$
64	0.0	2.6	1.3
121	4.1	3.4	2.7
196	1.7	3.4	0.4
289	4.3	4.0	2.6
400	3.1	3.3	0.6
Mean	2.6%	3.3%	1.5%

Table 3.3: Tour-length performance data for locally adaptive step-size methods. The data items are the tour percentage excess values. Runs using FA had settings of $\eta_+ = 1.2$ and $\eta_- = 0.5$, the SA runs settings of $\theta_+ = 1.05$ and $\theta_- = 0.95$. $K_r = .95$; other parameters as in the Table 3.1 BSD runs. When updating the $\{\eta_{jl}\}$ at the start of each epoch, the previous partial derivative terms are regarded as being zero. See main text for definitions of FA, SA and TLA.

to keep the system stable. Exploratory runs employing the latter idea show a slight improvement when resetting to 2.0 but a sharp deterioration when the reset value is pushed to 5.0: in such a high η regime the adaptation mechanism is overburdened, rather than accelerating convergence it is spending the high K iterations merely trying to prevent instability. Searching for the optimal (fixed) reset value using such semi ad-hoc methods is evidently a problem in the FA method.

Secondly, resetting the $\{\eta_{jl}\}$ at the start of every epoch discards potentially valuable information about the energy landscape built up during the previous epoch. For example, in a particular dimension, if the system spent the whole epoch moving in the same direction towards, but never reaching, a minimum, this information would be reflected in the associated η_{jl} variable having maximal value (through (3.3)). The problem is how — or whether — to utilise this information, given the complication that the energy landscape itself changes between epochs. The FA reset method denotes one extremal response to this

problem. The other is simply to let the adaptation scheme in (3.3) proceed unhindered by epoch boundaries, ignoring the problem mentioned earlier that the optimal step-sizes will likely be mismatched across an epoch boundary. Simulations letting the step-sizes adapt continuously in this manner do give slightly better results than FA for grid TSPs up to 289 cities, but unfortunately become unstable on the 400 city instance (data not shown).

3.3.1 The Twin Level Adaptation Method

Evidently, the source of the drawbacks just discussed in the FA method is the fact that the energy function changes during annealing, so that we are actually performing a series of adaptive minimization procedures (one at each K value) using the final state in one as the initial state in the next. This is a more complex situation than that encountered in the earlier adaptive step-size backpropagation studies, where adaptation occurred in the context of a constant energy surface. The natural improvement is to allow adaptation on two levels: fast adaptation within a particular epoch (as in FA), plus slow adaptation (SA) of a step-size scaling factor once per epoch. This twin level adaptation (TLA) strategy involves generalizing the (3.2) dynamics to

$$\Delta l_j^{(n)} = -K\theta_{jl}^{(k)}\eta_{jl}^{(n)}\left(\frac{\partial E}{\partial l_j}\right)^{(n)} \quad \forall j, l \tag{3.4}$$

where, as before, the $\{\eta_{jl}\}$ adapt according to (3.3) and are all reset to one at the start of each epoch. The superscript k labels the epoch number. Given that the η terms evolve using partial derivative sign information, this was also considered appropriate in the θ adaptation rule, detailed below:

$$\theta_{jl}^{(k)} = \begin{cases} \theta_+\theta_{jl}^{(k-1)} & \text{if } \left(\frac{\partial E}{\partial l_j}\right) \text{ has constant sign in epoch (k-1).} \\ \theta_-\theta_{jl}^{(k-1)} & \text{if } \left(\frac{\partial E}{\partial l_j}\right) \text{ changes sign in epoch (k-1).} \end{cases} \quad \forall j, l \tag{3.5}$$

The $\theta_{+/-}$ terms are analogues of the $\eta_{+/-}$ terms, and $\theta_{jl}^{(0)} = 1.0\,\forall j, l$.

As shown in Table 3.3, TLA performs better than FA alone, giving a mean percentage excess almost half that found by FA. To check whether the TLA performance is perhaps due solely to the slow θ adaptation, control runs using SA on its own (i.e. using the above dynamics but clamping the $\{\eta_{jl}\}$ terms to 1.0 throughout) were performed: the data, also shown in Table 3.3, strongly suggests that this is not the case. Thus, though either FA or SA will outperform BSD, a combination of the two strategies, TLA, is even more effective than either one on its own. Although the absolute differences in performance (characterised here by the mean percentage excess) between these methods may appear small, it is vital to remember that the arena in which successful TSP algorithms compete against one another is typically the set of tours possessing lengths within about 10% of the optimal length.[3] Finally, note that for comparable K_r, TLA also beats the energy monitoring scheme discussed earlier, even though the latter utilises global information, unlike the former.

These runs used values of 0.95 and 1.05 for θ_- and θ_+ respectively, with $K_r = .95$, and 10 updates per epoch. A higher value of K_r would produce less change in the energy surface per epoch, which would suggest that the θ variables need only change by a smaller amount than before, and hence that suitable $\theta_{+/-}$ values will be K_r-dependent. This intuition is supported by the finding that TLA runs still employing the above $\theta_{+/-}$ values but with a slower cooling schedule ($K_r = .99$), develop some very large θ values, oscillate and become unstable. To avoid resorting to a trial-and-error approach for finding suitable $\theta_{+/-}$, we would like to have a mathematical rule for setting these rates as a function of K_r. The following section provides this.

[3] As mentioned in section 1.4, even fairly naïve algorithms can find 20% suboptimal tours.

Stability Constraints on the θ Adaptation Rates

For the case of global η steepest descent dynamics, it was shown in section 3.1.1 that at high K the system would be stable against oscillations, provided $\eta\beta K < 0.5$. The high K requirement, recall, implied that the city forces were constant over a small region of the net and so could be ignored in exploring the dynamics within that region. Another interesting scenario concerns those beads lying between cities at low K values: as they have negligible interaction with the cities, their dynamics are also driven purely by tension forces. However, as a typical net is constructed with only two to three times as many beads as cities, the analysis should now assume that the bead in question has neighbour(s) closely bound to cities. Remaining for the moment with the simple case of global η dynamics, Figure 3.2a shows that a bead with both neighbours bound to cities will be stable to transverse fluctuations in the low K limit, provided that $\eta\beta K < 1.0$. A similarly simple calculation shows that this is also the stability condition with respect to lateral fluctuations.

To summarize, in the high K and low K regimes i.e., at the beginning and end of the annealing process, simplifications in the dynamics allow one to deduce, for beads in certain situations, a stability condition of $\eta\beta K < \Omega$, where Ω is of $O(1)$. The intermediate K regime, in which a small group of beads interacts with a small number of cities, would be far harder to analyse. Despite the absence of any such analysis, it is reasonable to conjecture that the above condition might still be roughly valid for intermediate K. Furthermore, there is the additional complication that these results were derived assuming a global η, whereas in TLA each bead has its own pair of effective step-sizes. We shall ignore this complication here and simply suggest that the corresponding TLA stability requirement is essentially $\eta\theta\beta K < \Omega$, with the various indices on θ and η being implicitly understood. Thus stability can be guaranteed throughout a simulation by ensuring both that the initial parameter set respects the above

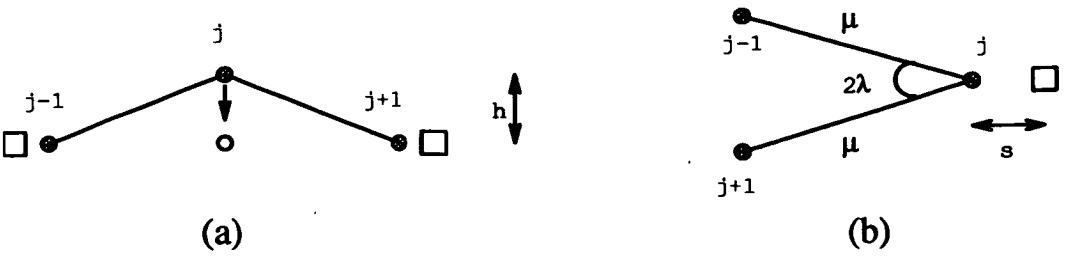


Figure 3.2: Low K scenarios. These sketches show different segments of the net at K low enough such that each city interacts only with the nearest bead. Squares denote cities, dots beads. (a) Shows bead j displaced from the equilibrium position (o). With global η dynamics, the restoring force will dampen the oscillation provided $2\eta\beta Kh < 2h$, i.e., if $\eta\beta K < 1.0$. (b) At equilibrium, the separation s equals $2\gamma K\mu\cos\lambda$, with γ and μ as defined in chapter 2. Assuming that the system reaches equilibrium by the end of an epoch, at the start of the next one, bead j will lie in a gradient field pushing it towards the new equilibrium position nearer the city. An approximate stability condition can then be derived by requiring the change in the bead's position during the first update, Δu, to be at most $O(\Delta s)$. For concreteness let us require $\Delta u < \Delta s$. With global η dynamics — and taking the simple case in which beads $j \pm 1$ remain fixed relative to bead j — this is equivalent to the inequality: $\eta < 1/\alpha$.

---

condition, and that the θ variables can not grow at a rate faster than that at which K decreases, i.e., max $\theta_+ = 1/K_r$. This result is in accord with the earlier discussion: it formalises the idea that gentler cooling should be accompanied by gentler θ adaptation. As K_r is typically close to 1.0, $1/K_r \approx (2 - K_r)$, and this is the θ_+ expression used in all subsequent TLA simulations. θ_- was chosen — somewhat arbitrarily — equal to K_r. In summary then, in the TLA method the adaptation rates θ_+ and θ_- are given by

$$\theta_+ = 2 - K_r, \qquad \theta_- = K_r. \tag{3.6}$$

Before presenting TLA data gathered using this rule, it should be noted that there is a class of beads for which the above restriction on θ_+ is insufficient to explicitly guarantee stability. These are the beads 'captured' by the cities at low K. As shown in Figure 3.2b, these beads need a local step-size condition $\eta\theta < 1/\alpha$ for stability. Now, for parameter sets following the minimal defect

prescription, $\beta/\alpha = 1/2\mu$, and respecting (3.6), one can calculate that in the low K regime (roughly, $.01\mu < K \ll K_c$), $1/\alpha < 1/\beta K$, so justifying the earlier statement about the θ_+ rule not guaranteeing stability. The cost of explicitly guaranteeing stability for both classes of bead, via a more stringent θ_+ rule, would be a narrowing of the potential dynamic range of the θ variables for beads in the previous class. This was considered unacceptable, for two reasons. Firstly, the θ values of such beads generated in simulations show that they *do* exploit the dynamical range allowed under the θ_+ rule above, so this presumably contributes to the efficacy of TLA. Secondly, the $1/\alpha$ condition was derived assuming K sufficiently low that each city has already captured a single bead; but by this stage the tour decision is complete. Of more importance is the high and intermediate K behaviour, for which the $1/\alpha$ condition is inapplicable.

Thus TLA à la (3.6) explicitly guarantees stability for those beads between cities, but relies on the ability of the adaptation mechanisms to maintain stable evolution of those beads nearest to cities. In practice this seems to be a successful policy. No stability problems were encountered with grid TSPs over a wide range of K_r. Furthermore, for all the grid TSPs, scrutiny of the mean final θ values of beads bound to cities showed them to be similar to, but always below, $1/\alpha$; indicating that these beads' step-sizes had adaptively tuned themselves to the maximum safe value.

3.3.2 Performance vs. Cooling Rate for Lattice TSPs

To compare the original steepest descent dynamics (BSD) with the best of the locally adaptive step-size techniques, namely TLA, the tour-length performance of both minimization methods was assessed on the standard set of lattice problems for several cooling rates, ranging from a very fast to an extremely slow cool. The results are shown in Figure 3.3. The cooling rate measure used is $1/\tau$, where τ denotes the number of epochs required for K to fall by an order of

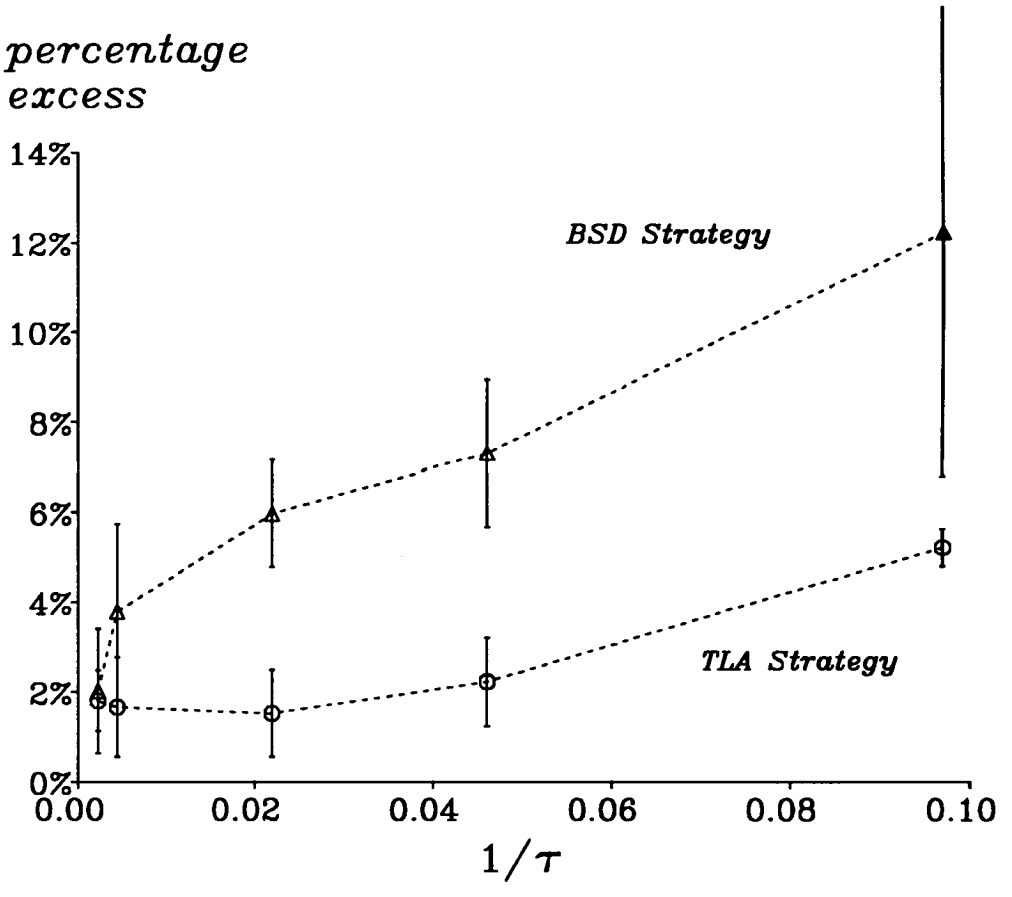


Figure 3.3: Plot of performance against cooling rate, comparing the BSD and TLA techniques on lattice TSPs. The ordinate denotes the tour percentage excess averaged over the five grid sizes used throughout this chapter, error bars the associated standard deviations. With the exception of one data point (filled), none of the runs generated spike or frozen bead defects. For the exceptional point, four out of the five grids required post-processing to remove frozen bead(s). BSD is defined in section 3.2, TLA by equations 3.4–3.6, with $\eta_+ = 1.2$ and $\eta_- = 0.5$. The other technical details are as listed in the legend to Table 3.1.

---

magnitude, i.e. $\tau \equiv -1/\log_{10} K_r$. There are several features to note here:

- BSD is unable to avoid generating nets with frozen bead defects when the cooling is very rapid, despite γ being set in accordance with the analysis of

the previous chapter. The size of the error bar on the $1/\tau = .097$ point is due to systematic growth in the number of frozen beads (with a corresponding decrease in post-processed tour-length quality) as the number of cities increases.

- Throughout the range of practicable cooling rates TLA produces shorter tours than BSD. In other words, TLA finds tours of a particular quality in less computer time than BSD; e.g. TLA is an order of magnitude faster than BSD in finding 5% sub-optimal tours (averaged over the set of grids).

- Only in the limit of extremely slow cooling is the BSD performance comparable to that of TLA. In this regime an adaptive step-size minimization method is unnecessary, as the changes in the energy surface are so gradual that even steepest descent has time to converge properly at each value of K. Notice however, that even in this limit typically the elastic net algorithm fails to find tours of optimal length — only for the $N = 64$ grid does the algorithm manage this. This point is considered further in the next section.

3.4 Performance on Random Euclidean TSPs

Although none of the techniques discussed so far in this chapter was explicitly dependent on the city distribution, all of the numerical explorations were conducted on regular lattice TSPs. It is therefore sensible to check whether the main conclusions also hold for other, harder, types of Euclidean TSP. This is the purpose of the current section.

TSPs in which the cities are scattered randomly within the unit square were studied, with six instances each for $N = 50$ and $N = 200$. In the 50 city group, the first five sets are those studied by [Durbin & Willshaw 1987] and

the sixth is from [Peterson 1990] (as is the sixth 200 city set). In order to estimate the optimal tour-lengths, the Or*-opt algorithm was run fifty times on each TSP using a different random initial tour each trial. The best final tour in each case was then regarded as 'optimal'. These tours agree with the best-known tours from previous studies (referenced below) for five of the $N = 50$ sets. For the remaining set plus the one $N = 200$ set previously studied, the procedure yielded tour-lengths within 1% of the best-known (from [Durbin & Willshaw 1987, Angéniol *et al.* 1988, Peterson 1990] and Frean (personal communication)).

3.4.1 Performance vs. Cooling Rate

Of the various adaptive minimization techniques, we shall consider only the final TLA strategy, as this looked the most promising on grid problems, and compare it against BSD. Simulations analogous to those in Figure 3.3 were conducted. The results are displayed in Figure 3.4.

The trends seen in Figure 3.4 are generally consistent with those seen for the grids. TLA gives better tours over most cooling rates, is far less prone to produce nets with frozen bead defects, and is only matched by steepest descent in the limit of slow cooling. Furthermore, the disparity in performance between BSD and TLA grows with the problem size. Regarding the average tour percentage excess over 'optimal', the slow cooling values of 3% and 5% for $N = 50$ and 200 respectively, are lower than those obtained on comparable problems by any of the eleven conventional tour construction heuristics evaluated in a recent review paper [Johnson 1990], though higher than those for decent iterative improvement algorithms such as Lin-Kernighan.

This raises the issue of what fundamentally limits the capability of the elastic net algorithm to find short tours. We saw in chapter 2 the importance of choosing the

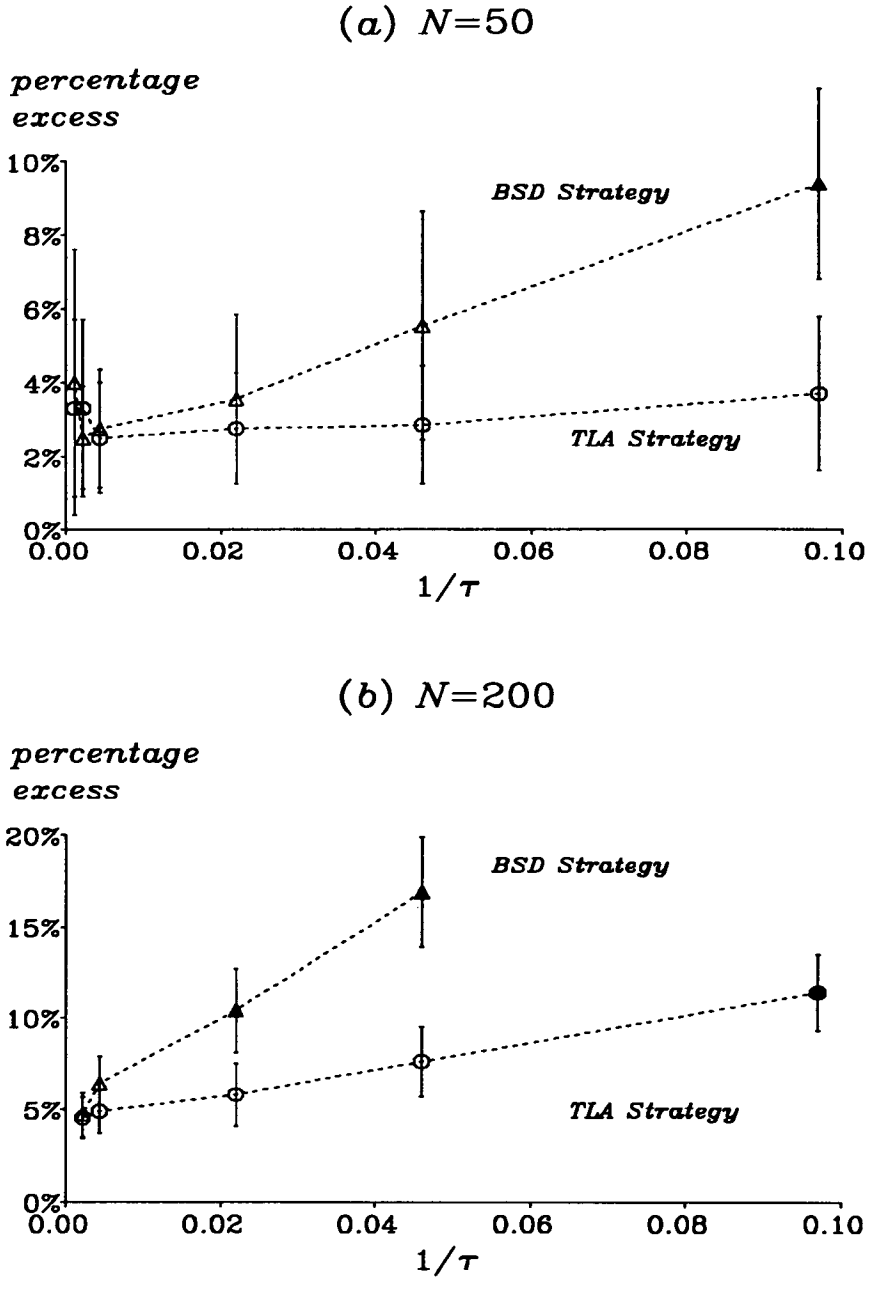


Figure 3.4: Plot of performance against cooling rate, comparing the BSD and TLA techniques on random TSPs. The ordinate denotes the tour percentage excess averaged over six TSP instances, error bars the associated standard deviations. Filled symbols denote points where three or more of the instances produced frozen beads. The $1/\tau = .097$ BSD data point lies off the scale and has been omitted. In graph (a) the deterioration at very low τ^{-1} may well be spurious, as in both cases it is due purely to diminished performance on only one of the six instances. See the Figure 3.3 legend for definition of the BSD and TLA techniques. Other technical details: $M = 2.5N$, $\beta/\alpha = \gamma'(= 2MN^{-1/2}/3)$, $\beta = 1.0$, initial state being a ring of radius .05 and $K = K_c$, ten steps per epoch.

key ratio β/α correctly, and in the current chapter how an efficient minimization strategy can boost performance through accelerating convergence to energy minima. However, it will be argued here that even if the above two issues can be dealt with satisfactorily, there is another factor, namely the deterministic nature of the algorithm, which limits its capabilities. Recall that the scale factor K in the energy function (2.3) acts to give a smooth energy landscape with few minima at high K, and a rugged landscape with many minima — some related to tours, others not — at low K. The algorithm operates by trying to find a deep minimum at high K and tracking it as K decreases. There is however no guarantee that the global energy minimum at any particular value of K will be accessible by descent dynamics from the region of phase space containing the global minimum at some higher value of K. Once in a certain minimum (or 'basin of attraction'), the system is bound to remain in that minimum, or one of its descendants through bifurcation, by virtue of the deterministic descent dynamics. Thus, it is argued that if the system enters an inappropriate basin early on, then no amount of slow cooling or parameter tuning will be able to compensate for this. Informal support for this claim can be found in Figure 3.5, which compares the best elastic net tour against the 'optimal' tour, for the six 50 city instances. Notice that the least optimal elastic net tours (those for instances 4 and 5, numbering from the left) show substantial differences in gross scale structure from their 'optimal' counterparts, presumably traceable to the high K period.

3.4.2 The Stochastic Elastic Net

It is perhaps not surprising that the deterministic nature of the standard elastic net algorithm causes problems, given that many of the other successful TSP heuristics e.g. edge-exchange algorithms, simulated annealing, genetic and self-organizing map algorithms, all make use of random variables. Randomness

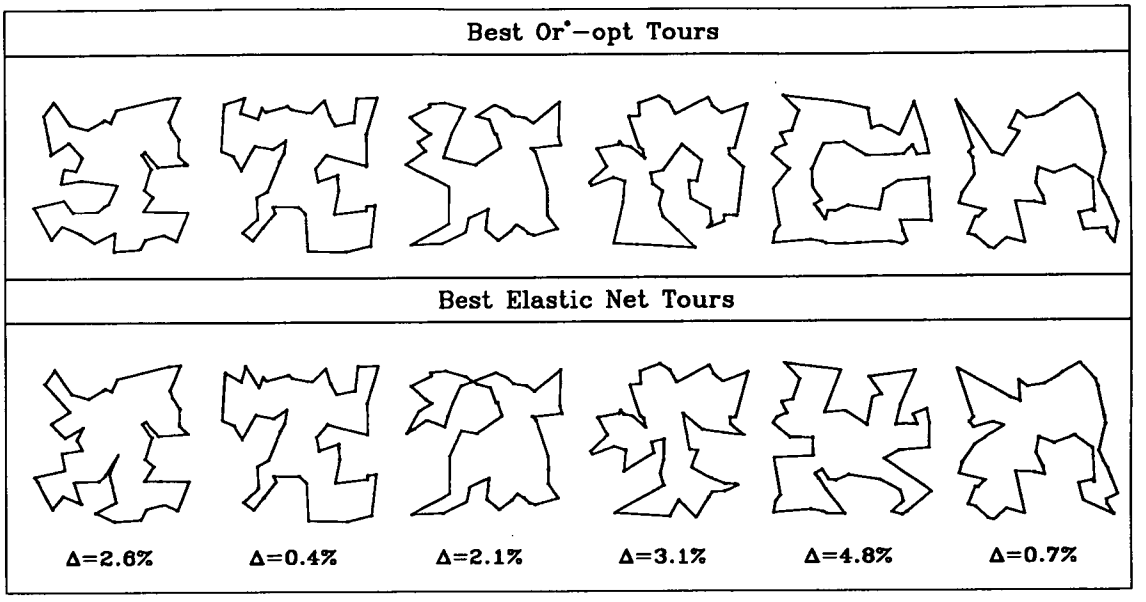


Figure 3.5: Comparison of the best 50 city tours. The upper row of tours are those regarded as quasi-optimal in this study. The lower row displays the corresponding best extrapolated tours found in the elastic net simulations summarized in Figure 3.4. Δ is the percentage by which these tours exceed the quasi-optimal tours in length.

---

can be utilised to allow variation in the initial state and, more powerfully, as an intrinsic component in the search dynamics. In both cases it aids the exploration of different regions of tour space. It is possible to incorporate randomness of the former variety into the elastic net method, simply by varying the initial configuration of beads. For example, rather than initializing the beads in a small ring centred on the cities' centroid (as in this study so far), the ring centre could be randomly displaced from the centroid. However, in practice, repeated runs using this initialization generated very little variation in the final tour. Although other schemes might be marginally more productive, the general policy of employing initialization noise will probably be ineffectual for the elastic net. The reason is that if the initial value of K is K_c or greater, the

dominant energy minimum in which the beads all lie at the centroid of the cities has a large basin of attraction, so that many initially different states flow towards the same initial minimum, thereby dampening the effect of the noise.

Adding Dynamic Noise

In light of the above comments, we shall now turn to the use of randomness within the annealing process itself. It is instructive to examine first how noise is utilised in self-organizing map TSP algorithms, for example those of [Angéniol *et al.* 1988] and [Fritzke & Wilke 1991], as these also operate geometrically by adapting a loop of beads towards the cities. In these algorithms, the basic step involves selecting a single city at random and then moving the section of the loop closest to that city even closer to it. The city selection order is random, and can be either quenched or dynamic. Either way the key point is that noise affects the evolution of the network throughout a run, enabling multiple runs to sample different regions of phase space.

This contrasts with the elastic net method, in which at every step the net effectively 'sees' all of the cities. A natural first step is therefore to present only some randomly selected fraction of the cities at each step. This, however, leads to highly unstable behaviour at low K. This problem arises because the elastic net has an explicit tension term to minimize the net length: the equilibrium location of the bead nearest to a particular city is determined by the balance between the tension force from its neighbours and the attraction of the city. If the city's influence is suddenly removed, the unbalanced tension force leads to a violent change in the bead's position. By contrast, the self-organizing methods rely on a neighbourhood adaptation mechanism to minimize the net length.

A more sensitive approach to varying the influence of the cities, particularly at low K, is required. This can be achieved by replacing, in the update equations,

the true weights by *auxiliary weights*, $\{w'_{ij}\}$, defined by

$$w'_{ij} = w_{ij}(1 + \xi_i) \tag{3.7}$$

where the ξ_i are independent random variables drawn from a Gaussian distribution of mean 0 and variance σ^2. The ξ_i are held fixed during an epoch but vary between epochs. The dynamical rules discussed before now perform descent, not on the energy function itself but on a randomly distorted version of it, E'.

$$E' = -\alpha K \sum_i (1 + \xi_i) \ln \sum_j \phi(|\boldsymbol{x}_i - \boldsymbol{y}_j|, K) + \frac{\beta}{2} \sum_j |\boldsymbol{y}_{j+1} - \boldsymbol{y}_j|^2. \tag{3.8}$$

This enables the system to 'climb over' barriers probabilistically in the true energy landscape. Regarding the setting of σ, the earlier problems of instability at low K suggest that σ ought to vanish as $K \to 0$. To accomplish this and still allow exploration early on in the high K region, σ can be set high initially and then decreased exponentially, as with K in the elastic net and temperature in simulated annealing.

A limited number of simulations were performed to test the feasibility of this 'stochastic elastic net' method. For each of the twelve testbed random TSPs, 20 runs were undertaken using quick cooling ($K_r = .95$), so that the total run time roughly matched that for a single very slow cool ($K_r = .9975$). The noise level σ was initially set to 5, then decreased at the same rate as K. TLA minimization was used, with the other parameters as in Figure 3.4. Calculating the percentage excess of the best run in each ensemble and then averaging over the six TSPs for each N value gives an aggregate measure of tour percentage excess for each problem size. This can then be compared with the corresponding figure for the best tours found in the Figure 3.4 experiments. For the $N = 50$ suite these figures are 1.7% and 2.2% respectively; for $N = 200$: 3.8% and 4.4% respectively. So adding dynamic noise can improve the performance of the elastic net. Further work optimizing the noise level and how this decreases, plus early pruning out of runs driven by the noise into poor regions of the energy landscape, would likely yield more gains.

In summary, the problem of determinism in the original elastic net method can be overcome by incorporating noise in a controlled fashion into the dynamics.

3.5 Conclusions

This chapter has focused on defining dynamics for the elastic net that allow it to converge rapidly into minima in the energy landscape. After a discussion on the problems of the original steepest descent dynamics, various adaptive descent strategies were proposed and investigated. The most successful strategies were those employing local step-sizes which adapt on the basis of changes in the partial derivatives. Such methods were first developed for backpropagation networks, in which the energy function remains constant. As the elastic net's energy landscape changes throughout a run, the original scheme was extended to allow adaptation on two timescales to exploit fully the power of the local step-size adaptation idea. Analytical work constraining the range of safe adaptation rates was presented. Extensive numerical experiments on two classes of TSP demonstrated that, for practicable run times, the new minimization technique generates shorter tours and fewer defects than the original steepest descent technique. Finally, problems due to the deterministic nature of the elastic net approach were discussed and a new 'stochastic elastic net' method proposed to ameliorate them. Exploratory experiments show the method to have potential.

Although developed in the context of the TSP, these local acceleration techniques could also be used to enhance the efficiency of the elastic net method in other applications, for instance in the cortical development model proposed in [Goodhill & Willshaw 1990, Goodhill 1992]. On a wider level, the local TLA heuristic could be employed to aid convergence on energy functions which vary regularly over time, such as those in mean field annealing algorithms.

CHAPTER 4

Potts TSP Network Dynamics

This and the following chapter study the mean field Potts algorithm for the TSP, introduced in section 1.5.3. The chapter is organized as follows. Section 4.1 summarizes and discusses the formalism of the method as originally introduced by [Peterson & Söderberg 1989] (PS hereafter), before describing in section 4.1.3 the general step-size dynamics which the algorithm employs in the current work. Sections 4.2–4.4 analyse the parameter dependence of an important characteristic temperature (T_c) under the new dynamics, for both parallel and serial updating modes. After accounting for the influence of certain secondary parameters, the theoretical T_c expressions are found to be in good agreement with the trends found in numerical experiments. The functional form of T_c is shown to strongly constrain the choice of suitable step-size values for parallel updating dynamics.

4.1 Formalism

As usual, we let N denote the number of cities and $\{d_{ij}\}$ the NxN matrix of intercity distances. The TSP is first mapped onto a spin system, through a matrix

representation scheme. Each city i is associated with N spin (decision) variables $S_{ia} \in \{0,1\}, 1 \leq i \leq N, 1 \leq a \leq N$. The index a labels the position, or partition number, in the list of cities defining a tour; the latter terminology arising from the the TSP's close relationship with the graph partitioning problem. The condition $S_{ia} = 1$ represents the decision that the ath position in the tour is occupied by city i. Conversely, if S_{ia} is 'off', i.e. zero, city i does not occupy position a. Thus the spin matrix matrix represents a tour if and only if it is a permutation matrix, i.e. iff $\sum_i S_{ia} = 1, \sum_a S_{ia} = 1, \forall\, i, a$.

4.1.1 Energy Functions

The spin system is defined by a quadratic energy function containing penalty terms which attempt to restrain the system into a tour state, and a cost term which yields the solution cost (here tour-length) when the state is a valid solution. One such energy mentioned in PS is $E^{spin}_{Ising} = E^{spin}_{t} + E^{spin}_{rest}$, where

$$E^{spin}_{t} \equiv \sum_{ij} d_{ij} \sum_{a} S_{ia} S_{j,(a+1 \bmod N)} \tag{4.1}$$

$$E^{spin}_{rest} \equiv \frac{\alpha}{2} \sum_{a} (\sum_{i} S_{ia} - 1)^2 + \frac{\beta}{2} \sum_{ib} \sum_{a \neq b} S_{ia} S_{ib} \;\geq 0 \tag{4.2}$$

Clearly E^{spin}_{rest} is zero if and only if the matrix represents a tour. Rather than relying on a restraint term to ensure that every city is 'on' exactly once, as above, PS explicitly constrained the network to do this, by enforcing the condition

$$\sum_{a} S_{ia} = 1 \quad \forall\, i. \tag{4.3}$$

Each city i now has an associated spin vector, $\boldsymbol{S}_i = (S_{i1}, \ldots S_{iN})$ constrained to be one of the principal unit vectors, hence the analogy (pointed out in PS) to the Potts model [Wu 1982] in statistical mechanics. Under this Potts condition, E^{spin}_{rest} is strictly equal to $\frac{\alpha}{2}(\sum_a (\sum_i S_{ia})^2 - N)$. However, by rewriting the second part as $\sum_{iab} S_{ia} S_{ib} - \sum_{ia} S^2_{ia}$, and enforcing (4.3) only on the $\sum_{iab}$ sum, E^{spin}_{Ising} transforms

to $E^{spin}_{Potts'}$ given (disregarding constant terms) by

$$E^{spin}_{Potts} \equiv \sum_{ij} d_{ij} \sum_{a} S_{ia} S_{j,(a+1 \bmod N)} + \frac{\alpha}{2} \sum_{a} (\sum_{i} S_{ia})^2 - \frac{\beta}{2} \sum_{ia} S^2_{ia} \qquad (4.4)$$

The reasons for retaining a term in β are discussed later in section 5.2.

4.1.2 The Mean Field Equations

The next step is to find the dominant states of the system defined by $E^{spin}_{Potts'}$ by studying the partition function $\mathcal{Z}$. The following remarks sketch the key steps in this procedure, see PS or the earlier work [Peterson & Anderson 1987] for more details. Regarding notation, $\boldsymbol{V}_i$ denotes the set $\{V_{ia} : 1 \leq a \leq N\}$, and unsubscripted $\boldsymbol{V}$ denotes the set $\{\boldsymbol{V}_i : 1 \leq i \leq N\}$ (similarly for $\boldsymbol{U}_i$ and $\boldsymbol{U}$). Rather than attempting an explicit summation over the N^N states which respect (4.3), PS first expressed $\mathcal{Z}$ as a multidimensional integral:

$$\mathcal{Z} = C \prod_{i} \int_{\mathcal{R}} d\boldsymbol{V}_i \int_{\mathcal{I}} d\boldsymbol{U}_i \, \exp\left(-E'(\boldsymbol{V}, \boldsymbol{U})\right) \qquad (4.5)$$

where

$$E' \equiv \frac{E_{Potts}(\boldsymbol{V})}{T} + \sum_{i} \left(\boldsymbol{U}_i . \boldsymbol{V}_i - \ln \sum_{a} e^{U_{ia}} \right), \qquad (4.6)$$

and

$$E_{Potts} \equiv \sum_{ij} d_{ij} \sum_{a} V_{ia} V_{j,(a+1 \bmod N)} + \frac{\alpha}{2} \sum_{a} (\sum_{i} V_{ia})^2 - \frac{\beta}{2} \sum_{ia} V^2_{ia}. \qquad (4.7)$$

Here, T is the temperature and C an unimportant constant. These equations can be derived by noting that a function $g(s)$, for real s, can be written as

$$g(s) = \frac{1}{2\pi i} \int_{\mathcal{R}} dv \int_{\mathcal{I}} du \, g(v) e^{u(s-v)} \qquad (4.8)$$

through δ-function manipulations, then generalizing to the case of $g = e^{-E_{Potts}/T}$. A *mean field* (MF) approximation is then made, which amounts to replacing the other spins by their average values when calculating the contribution to the integrand due to one particular spin. This allows the integrals to be decoupled, and

then evaluated by the saddlepoint method. The saddlepoint equations $\frac{\partial E'}{\partial V_{ia}} = 0$ and $\frac{\partial E'}{\partial U_{ia}} = 0$ generate a set of self-consistency equations in the MF variables:

$$\begin{aligned} U_{ia} &= -\frac{1}{T}\frac{\partial E_{Potts}}{\partial V_{ia}} \\ &= -\frac{1}{T}\left(\sum_j d_{ij}(V_{j,(a+1)} + V_{j,(a-1)}) + \alpha\sum_j V_{ja} - \beta V_{ia}\right), \\ V_{ia} &= \frac{\exp(U_{ia})}{\sum_b \exp(U_{ib})} \quad (\equiv f_{ia}). \end{aligned} \tag{4.9}$$

At a saddlepoint, denoted $(\widehat{\boldsymbol{U}}, \widehat{\boldsymbol{V}})$, one can identify TE' as an estimate of the free energy, $E_{Potts}(\widehat{\boldsymbol{V}})$ as the thermal average of E^{spin}_{Potts}, and identify the entropy $\mathcal{S}$ as

$$\mathcal{S} = -\sum_i \left(\widehat{\boldsymbol{U}}_i.\widehat{\boldsymbol{V}}_i - \ln\sum_a e^{\widehat{U_{ia}}}\right). \tag{4.10}$$

Henceforth E will refer to the quantity E_{Potts} (4.7), with E comprising components E_t, E_α and E_β (in the obvious decomposition). I shall also take the liberty of calling TE' the free energy (F), even at states which are not solutions of the mean field equations.

Observe from (4.9) that at a MF solution the variables respect a continuous version of the Potts condition (4.3), i.e.

$$\sum_a V_{ia} = 1 \quad \forall\, i \qquad \text{at a MF solution.} \tag{4.11}$$

For a related system [Peterson & Anderson 1987] showed that $\widehat{V_{ia}}$ was the expectation value of the spin S_{ia} within the MF approximation. Thus, given the normalization above, $\widehat{V_{ia}}$ can be interpreted naturally as the probability that city i is in the ath tour position. An alternative derivation of this result can be constructed by generalizing an analogous proof for the Ising model given in [Parisi 1988, Section 3.2]. This method also yields the mean field equations, though with a slightly different expression for U_{ia}; this is due to the diagonal 'self-interaction' terms being handled differently.

4.1.3 The Mean Field Annealing Algorithm

At high temperatures $F \approx -TS$, and so the mean field solutions will tend to be states near to the (symmetrical) maximum entropy state: $V_{ia} = 1/N \ \forall i, a$. Conversely at low T, $F \approx E$, so finding a MF solution will be equivalent to using a local optimization method on the internal energy — a procedure highly sensitive to the initial conditions and known to be ineffective [Hopfield & Tank 1985, p.149].

These characteristics are similar to those of simulated annealing, which is no surprise since both it and the MF method compute thermal averages over Gibbs distributions of discrete states, the former stochastically and the latter through a deterministic approximation. It is therefore natural to couple the mean field method with the concept of annealing from high to low temperatures, keeping to solutions of the MF equations at each intermediate temperature. The resulting *mean field annealing* method is a powerful and general technique which has found applications both in combinatorial optimization (e.g. for the GPP [van den Bout & Miller 1990]) and also in image restoration/edge detection (see, for example [Bilbro *et al.* 1992] and references therein).

In addition to the structure of the energy function, there are three major interdependent issues which arise in completely specifying a mean field annealing algorithm for an optimization problem:

- the values of the coefficients of terms in the energy function.
- the type of dynamics used to find solutions of the MF equations at each T.
- the annealing schedule details, i.e. the initial temperature (T_o), the rules for deciding when to reduce T and by how much, and the termination criteria.

Elements of all three issues will be addressed in this study. Before defining the dynamics, a few preliminary remarks concerning terminology and initial conditions are in order.

The symmetrical state, denoted by $\boldsymbol{V}^{sym}$, with components $V_{ia}^{sym} = 1/N \; \forall i, a$, is a trivial solution of the MF equations at all temperatures. In the high T phase $\boldsymbol{V}^{sym}$ is a stable fixed point. Below a critical temperature, T_c, this state turns unstable and the system undergoes a transition in which for each city i the V_{ia} variables begin to cluster, representing the emergence of the large scale structure of the tour. As the system is annealed below T_c the clustering process proceeds until (ideally) $\forall i, \exists a_i : \; V_{ia} = \delta_{aa_i}$ with $a_i = a_j$ iff $i = j$, indicating that each city has chosen a unique partition, i.e. the state unambiguously represents a tour. PS introduced a quantity called *saturation*, Σ, defined as

$$\Sigma = \frac{1}{N} \sum_{ia} V_{ia}^2 \tag{4.12}$$

to characterise the degree of clustering;[1] clearly $\Sigma_{\min} = 1/N$ and $\Sigma_{\max} = 1$. They also suggested, understandably, setting $T_o \approx T_c$ and the initial state to be near to, but displaced by some noise from, the symmetric state; the same will be done throughout most of this study.

4.1.4 Dynamics for Finding Mean Field Solutions

In PS, an iterative mapping method was used to evolve the V_{ia} variables towards a self-consistent solution of (4.9): at each iteration the V_{ia} were set equal to the corresponding f_{ia} terms, the procedure being stopped when the $|V_{ia} - f_{ia}|$ terms become 'small'. The cities could be updated either serially or in parallel. It is implicit in this method that the U_{ia} are essentially just auxiliary variables, computed as functions of the V_{ia} according to the solution form in (4.9) in order

[1] Other measures could also do this, e.g. an entropy measure $-\sum_{ia} V_{ia} \ln V_{ia}$, but Σ shall be used here for consistency with the original work.

to calculate the f_{ia} values. Notice that we could equally well define dynamics which evolve the U_{ia} variables, whilst treating the V_{ia} terms as auxiliary variables computed as functions of the U_{ia} via the solution form $V_{ia} = e^{U_{ia}} / \sum_b e^{U_{ib}}$. Such dynamics on U_{ia} will not be further elaborated on, except to say that in *neural* terms they correspond to updating the *potentials*, whilst V_{ia} dynamics refer to updating the *firing rates*.

In the current work, the iterative dynamics used in PS are generalized to the form:

$$V_{ia}^{(n+1)} = V_{ia}^{(n)} + \gamma(f_{ia} - V_{ia}^{(n)}), \quad 0 < \gamma \leq 1 \tag{4.13}$$

with the U_{ia} terms acting as auxiliaries, as above. Setting $\gamma = 1$ recovers the PS dynamics, whilst the $\gamma \to 0$ limit gives the first order Euler integration method for the differential equation $dV_{ia}/dt = f_{ia} - V_{ia}$, the fixed points of which constitute MF solutions. The reasons for studying these differential dynamics are threefold:

- contrary to the claims in the original paper, simulations (reported in section 4.4.3) employing parallel update $V_{ia}^{(n+1)} = f_{ia}$ dynamics do not give satisfactory performance. Specifically, the network converges into a low Σ state rather than a high Σ tour state. Similar problems in a mean field network for graph bisection were previously overcome by using the (4.13) dynamics with a small step-size γ [Peterson & Anderson 1988], though no analysis was presented to account for why this should help.

- as will be discussed later, the scaling behaviour of β implies that it is desirable for T_c to be as low as possible. Therefore one wants to understand how, if at all, T_c depends on γ.

- the differential equation solved by the continuous time limit of (4.13) is of a similar form to that which would govern the dynamics of a hardware electrical circuit designed to implement this algorithm. Technically,

the division operation required for the f_{ia} terms would make the circuit more complex than that for the Hopfield-Tank network discussed in section 1.5. Nevertheless, it is of interest to study the network using small γ, to examine the potential circuit behaviour.

One attractive feature of the $\gamma = 1$ dynamics, remarked in PS, is that the V_{ia} variables automatically satisfy the normalization in (4.11), even at states which are not solutions. A similar property holds under the general γ dynamics. To see this, form the sum over partitions a of both sides in equation 4.13 for any city i. This leads to

$$\Lambda_i^{(n+1)} = (1-\gamma)\Lambda_i^{(n)} + \gamma \tag{4.14}$$

where $\Lambda_i^{(n)}$ denotes $\sum_a V_{ia}$ at timestep n. Thus the V_{ia} remain normalized if they were normalized at the previous timestep. Furthermore, the dynamics ensure that any city having $\Lambda \neq 1$ will relax exponentially towards $\Lambda = 1$, since, writing $\Lambda^{(n)} = 1 - \epsilon^{(n)}$, it follows from (4.14) that $\epsilon^{(n+1)}/\epsilon^{(n)} = (1-\gamma)$. This is clearly a necessary property, as E, recall, was defined assuming Potts normalization.

Finally, there is the issue of whether the dynamics have an associated Lyapunov function in the continuous time limit. As the fixed points are, by construction, minima of the free energy function, it is natural to conjecture that F will be a Lyapunov function, as indeed it is for classic analogue neural networks [Hopfield 1984] employing potential dynamics. Skipping over the algebraic details, it can be shown that the continuous time dynamics, i.e. $dV_{ia}/dt = f_{ia} - V_{ia}$, imply

$$\frac{\mathrm{d}F}{\mathrm{d}t} = -T\sum_{ia}\frac{\mathrm{d}V_{ia}}{\mathrm{d}t}\frac{\mathrm{d}U_{ia}}{\mathrm{d}t} \tag{4.15}$$

It is not clear whether the r.h.s. here is always ≤ 0 (even after expanding the dU_{ia}/dt terms). Therefore it remains an open question whether the Lyapunov function is F or some other function, or indeed whether such a function exists for this system. Incidentally, U_{ia} dynamics also give (4.15), with expansion of dV_{ia}/dt similarly leading to no clearcut answer for the sign of dF/dt. However,

although the existence of a Lyapunov function is clearly a desirable property (as it gives some guarantee that the system will be 'well behaved'), it can be argued that for algorithms employing annealing it is not crucial. To understand why, first note that the locations of minima in the free energy landscape will typically only change by small amounts between two adjacent values of temperature in the annealing sequence. Thus, if the system has converged into one such minimum, it will only need to move a short distance through phase space in order to track the minimum. Relaxational dynamics will suffice for this task.

This contrasts with analogue Hopfield networks, where the system evolves along the whole trajectory from the initial state to a vertex, at a single temperature. In this case the system only ever reaches a MF solution at the very end of the trajectory. It is therefore imperative to have some understanding of what the dynamics are doing *along* the trajectory: the knowledge that F is a Lyapunov function for these networks provides that.

Analysis was presented in PS which estimated T_c as a function of the $\{d_{ij}\}$ distribution and parameters α and β, for both parallel and serial updating versions of the $V_{ia} \mapsto f_{ia}$ dynamics. In the following two sections, the original T_c analysis is generalized to the (4.13) dynamics (for both the parallel and serial cases). The basic structure of the new analysis remains that employed by Peterson and Söderberg; introducing the γ parameter brings additional complexity to the analysis but does not alter its basic structure.

4.2 Parallel Updating $T_c(\beta, \gamma)$ Analysis

Using parallel updating, consider a state at iteration n, perturbed slightly from the symmetric fixed point:

$$V_{ia}^{(n)} = V_{ia}^{sym} + \varepsilon_{ia}^{(n)}, \quad |\varepsilon_{ia}^{(n)}| \ll 1 \; \forall i, a. \tag{4.16}$$

Our goal is to find the conditions under which the dynamics of (4.13) cause the fluctuation terms ε_{ia} to grow and hence for $\boldsymbol{V}^{sym}$ to become unstable. Using equations 4.13 and 4.16, $\varepsilon_{ia}^{(n+1)}$ can be expressed as

$$\begin{aligned} \varepsilon_{ia}^{(n+1)} &= V_{ia}^{(n+1)} - V_{ia}^{sym} \\ &= \varepsilon_{ia}^{(n)}(1-\gamma) - \gamma V_{ia}^{sym} + \gamma f_{ia} \end{aligned} \tag{4.17}$$

Performing a Taylor expansion of f_{ia} around $\boldsymbol{V}^{sym}$ we have, to first order

$$\varepsilon_{ia}^{(n+1)} = \varepsilon_{ia}^{(n)}(1-\gamma) + \gamma \sum_{j,b} \varepsilon_{jb}^{(n)} \left.\frac{\partial f_{ia}}{\partial V_{jb}}\right|_{\boldsymbol{V}^{sym}} \tag{4.18}$$

which, using (4.9), gives

$$\varepsilon_{ia}^{(n+1)} = \varepsilon_{ia}^{(n)}(1-\gamma) + \frac{\gamma}{NT}\left[-\sum_j d_{ij}(\varepsilon_{j,a+1}^{(n)} + \varepsilon_{j,a-1}^{(n)}) - \alpha \sum_j \varepsilon_{ja}^{(n)} + \beta \varepsilon_{ia}^{(n)}\right] \tag{4.19}$$

Following Peterson and Söderberg, this can be recast in terms of Fourier coefficients $\widehat{\varepsilon}_{ik} \equiv N^{-1/2} \sum_{a=1}^{N} e^{\frac{-2\pi i k a}{N}} \varepsilon_{ia}$, to give the matrix equation

$$\widehat{\boldsymbol{\varepsilon}}_k^{(n+1)} = \left((1-\gamma)I + \frac{\gamma}{NT} M^{(k)}\right) \widehat{\boldsymbol{\varepsilon}}_k^{(n)} \tag{4.20}$$

where the matrix $M^{(k)}$ is defined by $M_{ij}^{(k)} = -2\cos(2\pi k/N)\, d_{ij} - \alpha + \beta\delta_{ij}$, and I is the $N \times N$ identity matrix. Note here that superscripts on the fluctuation terms label the iteration number, whereas on the evolution matrices and associated eigenvalues they label Fourier modes. From (4.20), the evolution of the $\widehat{\varepsilon}_{ik}$ and hence the ε_{ia} variables depends upon the dominant eigenvalues of the symmetric matrix $M^{(k)}$. Letting $\theta_{\max}^k$ and $\theta_{\min}^k$ denote the extremal eigenvalues of $M^{(k)}$, and defining $\theta_{\max} = \max_{k\neq 0}(\theta_{\max}^k)$, $\theta_{\min} = \min_{k\neq 0}(\theta_{\min}^k)$,[2] we see that T_c, the temperature below which at least one of the Fourier fluctuation coefficients will grow, hence turning the symmetric state $\boldsymbol{V}^{sym}$ unstable, is given by

$$T_c = \frac{1}{N} \max\left(\theta_{\max}, \frac{\gamma}{\gamma - 2}\theta_{\min}\right). \tag{4.21}$$

The $\gamma\theta_{\min}/(\gamma-2)$ case here corresponds to the situation $\widehat{\boldsymbol{\varepsilon}}_{k'}^{(n+1)} = -\widehat{\boldsymbol{\varepsilon}}_{k'}^{(n)}$ at T_c, k' being the Fourier mode giving $\theta_{\min}$. If the transition does proceed through this

[2] As $\widehat{\varepsilon}_{i0} = N^{-1/2}\sum_a \varepsilon_{ia}$ and the normalization (4.11) implies $\sum_a \varepsilon_{ia} = 0$, clearly $\widehat{\varepsilon}_{i0} = 0$, $\forall i$, therefore the eigenvalues for mode $k = 0$ are irrelevant.

channel, then at T_c, $\epsilon_{ia}^{(n+1)} \approx -\epsilon_{ia}^{(n)}$; in other words the dynamics will cause the V_{ia} variables to oscillate about V_{ia}^{sym} on successive updates, rather than to evolve smoothly, as they do when the transition proceeds through $\theta_{\max}$. Showing how to use the parameter β to avoid this troublesome oscillatory transition was one of Peterson and Söderberg's major contributions. Their first step was to analyse the eigenvalue spectrum in order to estimate $\theta_{\max/\min}$ in terms of α, β and three parameters characterising the TSP instance, namely the number of cities N, plus the mean $\langle d \rangle$ and standard deviation (σ_d) of the off-diagonal elements in the $\{d_{ij}\}$ matrix. Their key results, summarized below, can be imported directly into the current work:

- By defining a matrix $A^{(k)}$ with elements

$$A_{ij}^{(k)} = g(k)d_{ij} - \alpha(1 - \delta_{ij}), \tag{4.22}$$

  where $g(k) = -2\cos(2\pi k/N)$, the matrix $M^{(k)}$ can be written

$$M^{(k)} = (\beta - \alpha)I + A^{(k)}. \tag{4.23}$$

  Thus

$$\theta_{\max/\min} = (\beta - \alpha) + \lambda_{\max/\min} \tag{4.24}$$

  where $\lambda_{\max} = \max_{k\neq 0}(\lambda_{\max}^k)$, $\lambda_{\min} = \min_{k\neq 0}(\lambda_{\min}^k)$, the λ^k being eigenvalues of $A^{(k)}$.

- Using perturbation analysis, to first order, $A^{(k)}$ has two distinct eigenvalues:
  - $\lambda_{\parallel}^k = (N-1)(g(k)d - \alpha)$, with eigenvector $(1,1,1,\cdots)$.
  - $\lambda_{\perp}^k = -(g(k)d - \alpha)$; this is an $(N-1)$ fold degenerate group, with eigenvectors orthogonal to $(1,1,1,\cdots)$.
- At second order there is a negligible correction to $\lambda_{\parallel}^k$ but the degeneracy of the $\lambda_{\perp}^k$ is lifted, with the eigenvalues spread over a range with

limits $-(g(k)d - \alpha) \pm g(k)\zeta\sqrt{N}\sigma_d$. ζ is a term which characterises the width of the $\lambda_\perp$ range, its value was found empirically by PS to be $\chi\sqrt{N}$ with $\chi \approx .65$ for uniformly random square distributions of cities.

- At this stage PS made implicit assumptions that $\lambda_{\min} = \min_{k\neq 0}(\lambda_\parallel^k)$ and $\lambda_{\max} = \max_{k\neq 0}(\lambda_\perp^k)$. Basic algebra shows that the first assumption is valid provided $d > \sigma_d\chi$ which, as will be discussed shortly, should be true for most reasonable $\{d_{ij}\}$. For large values of N, the second assumption is valid only when

$$\alpha > 2(d - \sigma_d\chi). \tag{4.25}$$

We will assume here that α is chosen to satisfy this inequality (the issue of setting α is dealt with more fully in section 5.1). So finally, the PS estimates for $\lambda_{\max}$ and $\lambda_{\min}$ are

$$\lambda_{\min} = -(N-1)(2d+\alpha), \qquad \lambda_{\max} = \alpha + 2d + 2\zeta\sqrt{N}\sigma_d. \tag{4.26}$$

Incorporating (4.24) into the current analysis through (4.21) gives:

$$T_c = \frac{1}{N}\max\left(\beta - \alpha + \lambda_{\max}, \frac{\gamma}{(\gamma-2)}(\beta - \alpha + \lambda_{\min})\right). \tag{4.27}$$

Noting that $\gamma/(\gamma-2) < 0$ for $0 < \gamma \leq 1$, the oscillatory transition can thus be avoided by setting $\beta : \beta > \beta_o$, where

$$\beta_o = \alpha - \frac{1}{2}\left[\gamma(\lambda_{\min} - \lambda_{\max}) + 2\lambda_{\max}\right]. \tag{4.28}$$

Simple algebra yields:[3]

$$T_c(\beta_o) = \frac{\gamma}{2N}(\lambda_{\max} - \lambda_{\min}) \tag{4.29}$$

$$T_c(\beta) = \begin{cases} T_c(\beta_o) + (\beta - \beta_o)/N & \text{if } \beta > \beta_o \\ T_c(\beta_o) + \frac{\beta-\beta_o}{N}\frac{\gamma}{\gamma-2} & \text{if } \beta \leq \beta_o. \end{cases} \tag{4.30}$$

Using the estimates in (4.26) gives β_o, in the large N limit, as

$$\beta_o = Nd\left[\gamma(\chi\sigma_d/d + 1 + \alpha/2d) - 2\chi\sigma_d/d\right]. \tag{4.31}$$

[3]Note that these equations reduce to the corresponding PS equations when $\gamma = 1$.

This is a key result, showing how for parallel updating, the sign of β_o depends on three factors:

1. The particular problem instance $\{d_{ij}\}$, through the product of σ_d/d and χ. σ_d/d is readily computable from the $\{d_{ij}\}$ matrix and can also be calculated analytically (in the $N \to \infty$ limit) for simple distributions. For example, a uniformly random distribution in a square box of side L has $d = 0.521L$ [Bonomi & Lutton 1984], and by a simple integral $\langle d_{ij}^2 \rangle = L^2/3$. Hence $\sigma_d/d = 0.478$ for this distribution. Rectangular distributions cause σ_d/d to rise, with the limiting 1-D case having $\sigma_d/d = 1/\sqrt{2}$. Calculations performed for model distributions in which the cities are equally divided amongst several distinct clusters suggest that σ_d/d is usually a constant (< 1) for this class too. The χ factor is less transparent. It is not clear which property of the city distribution it depends on; third order perturbation analysis may be one approach worth future exploration in this regard. However, *empirically*, the χ values for the three distributions discussed above are approximately 0.62, 0.8 and < 1 respectively, for large N.[4]

2. The value of α; ostensibly α is a free parameter, but, as will be discussed in section 5.1, the form of the energy function leads to $2d$ being a suitable value for α. Thus the value of α is determined by the problem instance.

3. The updating step-size, γ.

Thus for any given TSP, the only way to control β_o is through the parameter γ. If γ is fixed at 1.0, as in PS, then this degree of control is lost, with adverse consequences on the algorithm's performance: for the types of distributions discussed above, $\chi\sigma_d/d < 1$, so from equation 4.31 β_o will increase linearly with N. But with this scaling on β_o, and hence also on β as we require $\beta > \beta_o$,

[4]Distributions that contain a small number of very distant outlying cities have $\sigma_d/d \sim \sqrt{N}$ and $\zeta \sim N^\kappa$, $0 \leq \kappa \leq 1/2$ — such distributions will not be considered further.

the tour-length energy term will effectively be ignored, since the changes in energy in going from an initial $\Sigma_{\min}$ state to a final $\Sigma_{\max}$ state will be such that $\Delta E_\beta / \Delta E_t \sim N$. The likely result is either a poor quality tour, or a state which does not represent any valid tour at all. Thus a step-size value of 1 is inappropriate for dynamics updating the cities in parallel.

To avoid β being forced to scale with N, γ must be chosen so that $\beta_o < 0$. From (4.31) this can be done by choosing γ below a critical value, γ_c:

$$\gamma_c = \frac{4\chi\sigma_d}{2\chi\sigma_d + 2d + \alpha}. \tag{4.32}$$

The work so far has centred on how the choice of β is governed by two objectives: first, to keep β as low as possible so as to maximize the influence of the tour-length term in the energy function, and second, to satisfy $\beta > \beta_o$ so that the transition at T_c proceeds smoothly rather than through the oscillatory mode. The analysis so far implies that β may safely be allowed to fall to zero, provided that $\gamma < \gamma_c$. However, we shall see later that, irrespective of the updating rule's form, in order for low cost tours to be self-consistent solutions of the mean field equations, β must be a small but non-zero $O(1)$ term. Using (4.26) and (4.27) it is straightforward to check that this additional constraint on β does not alter the range of satisfactory γ values — γ_c still defines the upper limit of the range.

In summary, the problem with parallel updating using $V_{ia} = f_{ia}$, is that in order to ensure that the system evolves smoothly at the critical temperature, the coefficient β must be of $O(N)$, which destroys the algorithm's ability to find decent tours. With reference to (4.26) and (4.28), this arises fundamentally because, for the types of city distributions discussed earlier, the dominant eigenvalue over the set of matrices $A^{(k)}$ is $\lambda_{\min}$ rather than $\lambda_{\max}$, and β must be taken large to compensate for this. By moving to incremental dynamics of the form in equation 4.13 and taking γ small, the effect of the negative eigenvalue's dominance can be overcome, essentially by weighting the new V_{ia} values to be close to the old ones, rather than by using an unacceptably high β.

4.3 Serial Updating $T_c(\beta, \gamma)$ Analysis

Serial updating involves applying the dynamical rule of (4.13) to each city in turn until all have been updated. The order in which the cities are updated is random. Whether this order remains constant or varies throughout a run does not alter the T_c analysis. A fixed order is employed in the software implementation.

The early steps of the analysis, i.e. writing the full equation for the evolution of fluctuations around the symmetry point, linearizing it, and then transforming into Fourier space, are very similar to those in the parallel updating case, and so will not be detailed here. However, as the calculation of each $V_{ia}^{(n+1)}$ will in general involve using some variables from the previous iteration, plus those already updated during this sweep, the serial analogue of equation 4.20 is given by

$$\widehat{\varepsilon}_k^{(n+1)} = (1-\gamma)I\widehat{\varepsilon}_k^{(n)} + \frac{\gamma}{NT}\left(L^{(k)}\widehat{\varepsilon}_k^{(n+1)} + (U^{(k)} + D)\widehat{\varepsilon}_k^{(n)}\right) \tag{4.33}$$

where the cities have been relabelled to match the updating order. The matrices $U^{(k)}$, $L^{(k)}$ and D are respectively the upper triangular, lower triangular and diagonal components of the matrix $M^{(k)}$ used in the parallel updating analysis. Gathering like terms yields

$$(NTI - \gamma L^{(k)})\widehat{\varepsilon}_k^{(n+1)} = \left[NT(1-\gamma)I + \gamma(U^{(k)} + D)\right]\widehat{\varepsilon}_k^{(n)} \tag{4.34}$$

and hence

$$\widehat{\varepsilon}_k^{(n+1)} = W^{(k)}\widehat{\varepsilon}_k^{(n)}, \quad W^{(k)} \equiv (NTI - \gamma L^{(k)})^{-1}\left[NT(1-\gamma)I + \gamma(U^{(k)} + D)\right]. \tag{4.35}$$

$W^{(k)}$ is in general non-Hermitian, and will therefore possess complex eigenvalues. T_c is the temperature at which the first eigenvalue (in the set of $W^{(k)}$ matrices) touches the unit circle. Following the method of Peterson and Söderberg, consider the two distinct cases for an eigenvalue μ of unit modulus, namely $\mu = +1$ and $\mu = e^{2i\phi}, \phi \neq 0$.

In the former case we have $\widehat{\varepsilon}_k^{(n+1)} = \widehat{\varepsilon}_k^{(n)}$, which, upon substitution into (4.34), yields

$$\widehat{\varepsilon}_k^{(n)} = \left[(1-\gamma)I + \frac{\gamma}{NT}M^{(k)}\right]\widehat{\varepsilon}_k^{(n)} \tag{4.36}$$

implying that the matrix operator in this line has eigenvalue +1. But this is just the evolution operator found for the parallel updating case. Hence the $\mu = +1$ mode causes instability at $T = \theta_{\max}/N$.

Substituting $\mu = e^{2i\phi}$ into (4.34) and multiplying through by $e^{-i\phi}$, leads to

$$\gamma(e^{-i\phi}U^{(k)} + e^{i\phi}L^{(k)})\widehat{\varepsilon}_k^{(n)} = \left[NT(e^{i\phi} - e^{-i\phi}(1-\gamma)) - e^{-i\phi}\gamma(\beta-\alpha)\right] I\widehat{\varepsilon}_k^{(n)}. \tag{4.37}$$

The left hand matrix operator is Hermitian, and so the imaginary part of the right hand numerical term must vanish. This implies that $T = \gamma(\alpha-\beta)/(2N - N\gamma)$. Thus T_c for the case of serial updating is given by

$$T_c(\beta,\gamma) = \frac{1}{N}\max\left(\frac{\gamma(\alpha-\beta)}{2-\gamma}, (\beta - \alpha + \lambda_{\max})\right). \tag{4.38}$$

From this result one can go on to derive the serial versions of equations 4.28–4.30. The key point however, is that the types of city distributions discussed earlier have $\lambda_{\max} \sim N$, so β_o — the value of β above which the leading $W^{(k)}$ matrix has a dominant eigenvalue of +1 at T_c — is negative for all γ. Thus, for the β values of practical concern i.e. those > 0, T_c is $(\beta - \alpha + \lambda_{\max})/N$, as it also is for parallel updating, provided that $\gamma < \gamma_c$ in the latter case. Indeed it is straightforward to show that, in the limit of $\gamma \to 0$, the analytical T_c values for the serial and parallel updating cases become equivalent across the entire β range, for any city distribution. This provides a consistency check on the serial and parallel $T_c(\beta,\gamma)$ analyses: from (4.13) the change in any of the mean field variables during an update is $O(\gamma)$, thus in serial updating the effect of using some variables which have already been updated during this sweep, rather than their values at the previous sweep, becomes negligible as $\gamma \to 0$. Therefore the two updating modes ought to be equivalent in this limit.

In summary, for practical values of β, generalizing the dynamics to the form of (4.13) should have no effect on the critical temperature if the MF variables are updated serially. With parallel updating however, the situation becomes more interesting; for low values of β, T_c is expected to become independent of γ only if γ falls below a certain critical value. Because of the parallel case's greater complexity, the numerical experiments in the following section focus almost exclusively on this updating mode.

4.4 Numerical Tests of the $T_c(\beta, \gamma)$ Analysis

The foregoing analysis of the transition temperature's parameter dependence was tested through extensive numerical experimentation with a software implementation of the algorithm. This work was performed on a Meiko Computing Surface, a message-passing multicomputer system. Using either parallel, or (perhaps surprisingly) serial updating, the algorithm can be efficiently mapped onto this parallel architecture; the parallelization strategy is discussed in Appendix A.

Before presenting the $T_c(\beta, \gamma)$ data, some technical details of how the algorithm was actually implemented will be covered. This is an important — though often neglected — issue in neural networks research. The path from a theoretically clean model to a robust software implementation often involves the introduction of new parameters to specify or control secondary aspects of the algorithm, such as the initial conditions, convergence and termination criteria. Sensitivity to the values of these 'secondary' parameters complicates the task of interpreting the algorithm's behaviour as a function of its 'primary' parameters, *viz*, the coefficients of the terms in the energy function and the dynamical timestep. In the implementation of the Potts TSP algorithm used here there are several secondary parameters, the values of which can affect markedly both the

measured transition temperature and final performance. It is therefore clearly necessary to understand these issues, at least qualitatively, before one can make any sensible comparison with the analytical results.

4.4.1 Implementation Details

The general approach is to initialize the system close to the symmetric state at a temperature near T_c, then, allowing the system to converge to a fixed point at each temperature, anneal (i.e. reduce T) until either the system reaches a stable high saturation state or some other termination criterion is satisfied. Specifically, the implementation used here has the following ingredients:[5]

initial state $V_{ia}^{n=0} = (1 + \xi_{ia})/N$, with the ξ_{ia} being random variables drawn uniformly from $[-\xi, \xi]$. Clearly ξ should be $\ll 1$.

convergence At each temperature the V_{ia} variables are updated by the rule in equation 4.13 until they reach — to a certain tolerance — a fixed point, which, as discussed earlier, represents a free energy minimum. The degree of convergence is monitored through the quantity Δ^{expt}, denoting the sum $\sum_{ia} |V_{ia} - f_{ia}|$. If $\Delta^{expt} < \Delta'$, where $\Delta' \equiv \Delta(0.05/N)$, the system is deemed to have reached a fixed point and the temperature is reduced. Δ will henceforth be called the tolerance parameter.

annealing schedule A simple exponential scheme is used, i.e. after reaching a fixed point at T, $T \mapsto T \times T_r$, where $T_r \in (0, 1)$ controls the cooling rate. Roughly speaking, T_r should be set to at least 0.9.

termination criteria Three criteria are employed, the first being that the saturation Σ exceeds some threshold near to $\Sigma_{\max}$ at a fixed point; a threshold of 0.9 was used in this study. The others are designed to trap rogue runs, by

[5]These broadly follow the method used in PS but with some additions and improvements.

terminating runs which either fail to reach a fixed point within a certain number ($\mathcal{M}$) of sweeps at some temperature (a sweep being one update for every city), or which allow T to fall below $T_{\min}$, ($T_{\min} \ll T_c$), without ever satisfying the $\Sigma > \Sigma^{thresh}$ criterion. $T_{\min}$ was set to 10^{-3}.

tour extrapolation A robust procedure for generating a likely tour from any final $\{V_{ia}\}$ configuration was developed. First, for each city i, find the partition i' *visited* by that city, under the rule: $V_{ii'} = \max_a(V_{ia})$. If every partition has exactly one visit, then the mapping onto a city ordering, i.e. tour, is trivial. If this is not the case, then strictly speaking no tour is defined. In practice however, a tour can still be generated from such states through a modest amount of post-processing to order the cities which have visited a common partition. In the current implementation, for partitions visited by two cities the better of the two possible orderings is chosen. For partitions visited by three or more cities, a random order is chosen.

4.4.2 Influence of the Secondary Parameters on T_c

Sections 4.2 and 4.3 developed expressions for T_c, the value of the temperature below which the symmetric state turns unstable. This section demonstrates that the effect of the secondary parameters is typically to drive the experimental values of T_c below those predicted, and presents arguments to account for why this should happen.

Defining the experimental transition temperature, T_c^{expt}, to be the temperature at which Σ first exceeds $\Sigma_{\min}(= 1/N)$ by 1%, a series of runs were performed to investigate the dependence of T_c^{expt} on the tolerance (Δ), level of initialization noise (ξ), initial temperature (T_o), and cooling rate parameter (T_τ). Two TSPs with cities from a uniformly random unit-square distribution were studied, having $N = 50$ and $N = 200$. Ten trials using different initialization noise for

the $\{V_{ia}^{(n=0)}\}$ state were run at each parameter point; data points and their corresponding error bars in the graphs represent the mean and standard deviation values over the ten trials. These procedures were used for all the experiments throughout this chapter, unless stated otherwise. The runs reported in this section were performed using parallel updating dynamics.

The values of the primary parameters (α, β, γ) were fixed at $(1.0, 0.5, 0.2)$ and the values of the secondary parameters (Δ, ξ, T_o, T_r) varied. Figure 4.1 shows the variation of T_c^{expt} with Δ and ξ across a range of T_o values, for fixed T_r, at $N = 50$; the $N = 200$ data show similar trends. Temperatures are normalized to the predicted transition temperature, T_c^{theo}, calculated from equations 4.26 and 4.29–4.31, using the continuum values of d and σ_d (i.e. 0.52 and 0.25) and a χ value of 0.65, typical for TSPs of this size. The theoretical profiles in Figures 4.1 and 4.2 are constructed on the simple assumption that the transition is immediate for all T_o values up to T_c^{theo}, and occurs at T_c^{theo} for $T_o > T_c^{theo}$.

The main feature of Figure 4.1 is that, for $T_o \approx T_c^{theo}$, T_c^{expt} is usually substantially lower than T_c^{theo}, with the difference increasing as T_o/T_c^{theo} grows further. Furthermore, the T_c^{expt} values are higher, i.e. closer to the theoretical values, for high values of the initialization noise (ξ), or low tolerance (Δ). These results can be qualitatively explained by the following arguments. Consider the dynamics at $T_o \approx T_c^{theo}$, with low ξ and/or high Δ: the $\boldsymbol{V}^{sym}$ state is unstable, therefore the noise causes Δ^{expt} to grow (as the system moves away from that state). However, if Δ is so large that, despite its growth, Δ^{expt} at T_o is still quickly $< \Delta'$, then by the convergence criterion defined in section 4.4.1, the system will be deemed to be at a fixed point and T will be reduced. In this manner, T may fall quite far below T_c^{theo} before the system has time to move far enough away from the symmetric state to register a transition under the $\Sigma > 1.01/N$ rule. Thus T_c^{expt} will be lower than T_c^{theo}, even if the eigenvalue analysis which gave the T_c^{theo} expression is correct.

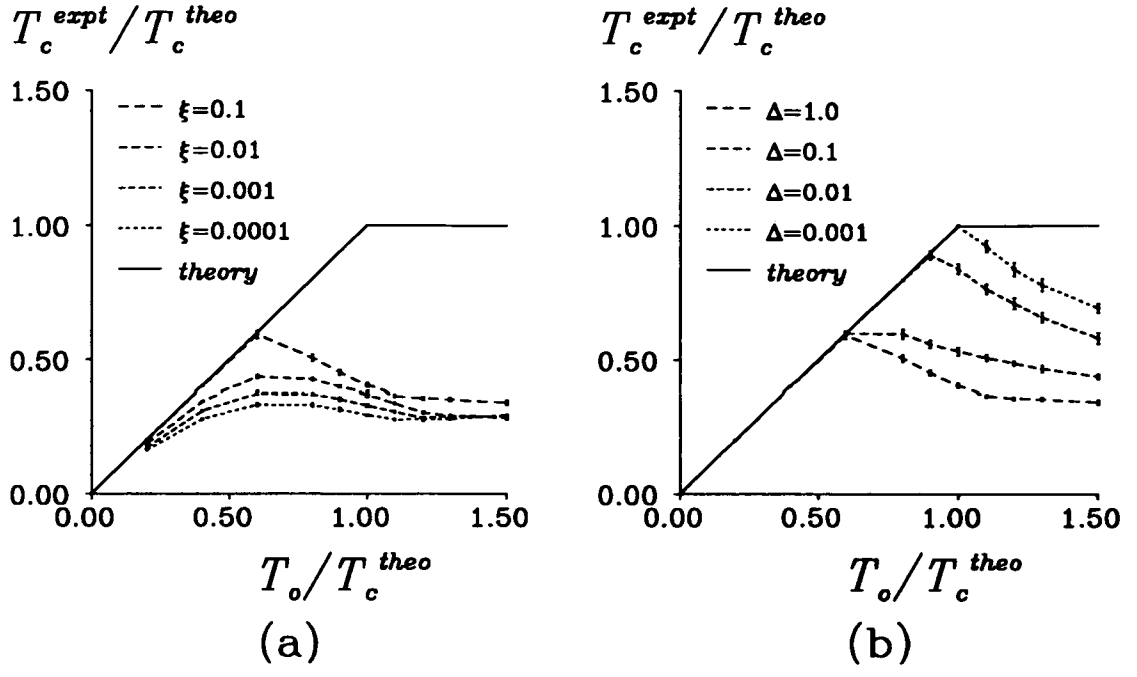


Figure 4.1: T_c^{expt} plotted against T_o, with both quantities normalized to T_c^{theo}; $T_r = 0.95$ for both plots. (a) collective plot showing the effect of varying ξ with Δ fixed at 1.0. (b) collective plot showing the effect of varying Δ with ξ fixed at 0.1.

Similarly, a very low ξ value for $T_o \approx T_c^{theo}$ requires many sweeps before the instability can grow to any extent. But unless Δ is commensurately low, the implementation rules will declare the fixed point reached (and hence reduce T) long before the system has had sufficient sweeps to move far away from the unstable equilibrium point at that T. The behaviour for $T_o > T_c^{theo}$ can also be understood in these terms: above T_c^{theo} the symmetric state is stable, so any initial noise will be reduced or even extinguished by the dynamics. Thus, once T has fallen to $\approx T_c^{theo}$ the effective noise level will be very small, and the argument above becomes applicable again.

In essence, this account argues that the T_c^{expt} values in Figure 4.1 are less than T_c^{theo} because the secondary parameters used did not allow the system sufficient sweeps to relax fully at temperatures close to T_c^{theo}. This phenomenon is reminiscent of the 'critical slowing down' observed in simulations of certain spin

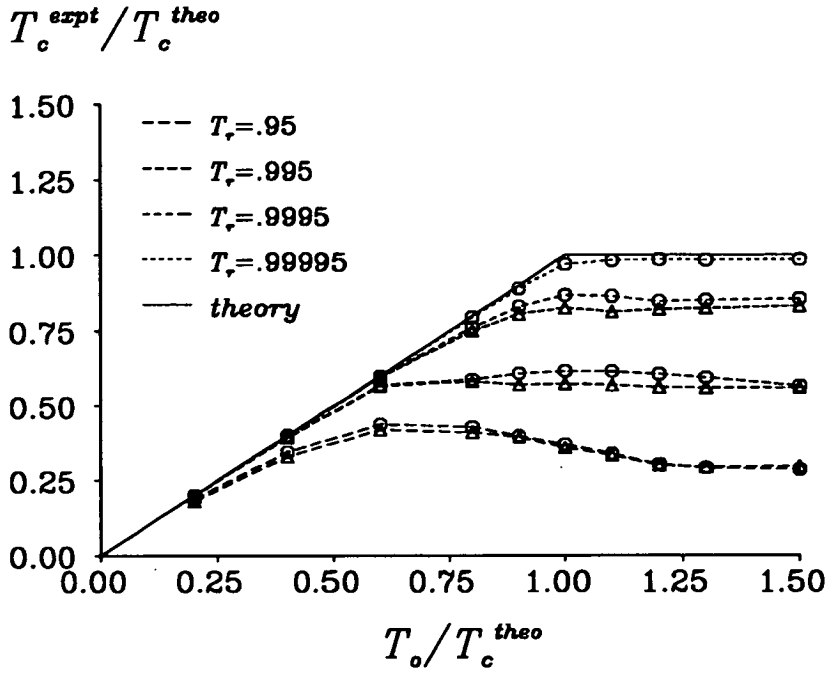


Figure 4.2: T_c^{expt} plotted against T_o, with both quantities normalized to T_c^{theo}. Collective plot showing the effect of varying the annealing rate, with $\Delta = 1$ and $\xi = 0.01$. Circles denote $N = 50$ data, triangles $N = 200$; the error bars are of order 0.01 or less, and so have been omitted for clarity.

systems, in which the equilibration times grow as a second-order phase transition is approached. Behind this similarity however, the underlying causes are different. In the spin systems this behaviour is due to divergence of the correlation length, whereas in the current system it results from the leading eigenvalue in (4.20) only just exceeding unity for T just below T_c, thereby requiring a large number of sweeps before the system is driven along the associated eigenvector and towards the new free energy minimum displaced from the symmetry point.

One simple way to test this idea is to anneal more slowly, thereby allowing the system more time to relax across all temperatures. Figure 4.2 shows the effect of varying the annealing rate (T_r): the T_c^{expt} values do indeed rise as T_r rises. Furthermore, notice the convergence to the theoretical profile as $T_r \to 1$ and the lack of any significant dependence on N in the data. These points are important, because the earlier interpretation of the Figure 4.1 data first *assumed*

the underlying T_c analysis to be correct and then argued why typical choices of the secondary parameters lead to a systematic reduction in T_c^{expt}. Figure 4.2 provides direct support for the veracity of the T_c analysis, albeit strictly only for this particular (α, β, γ) point. However, using extremely slow annealing and/or low Δ leads to unacceptably long run times. Hence in the experiments reported in section 4.4.3 checking the functional form of $T_c(\beta, \gamma)$, no attempt has been made to show convergence to the theoretical profile. Instead, practical settings of the secondary parameters were employed and the existence of a systematic reduction factor (for T_c) assumed.

Finally, it should be noted that the secondary parameter settings (particularly T_τ and Δ) not only complicate the issue of checking the T_c analysis, but they also have a strong influence on the quality of tours found by the algorithm. Since the mean field approach can be viewed as a deterministic form of simulated annealing, it is not surprising that, as in simulated annealing, the solution quality improves as $T_\tau \rightarrow 1$. Δ can also be viewed as a cooling schedule parameter, as it determines the number of sweeps performed at each temperature — in general the lower Δ is, the better the quality.[6] The interesting point, brought out in Figure 4.3, is that T_τ influences the quality chiefly through determining T_c^{expt}.

As T is lowered through T_c^{theo}, first one and then more eigenvalues of the $\{(1-\gamma)I + \frac{\gamma}{NT}M^{(k)}\}$ set of matrices in equation 4.20 exceed unity in magnitude, opening up eigenvectors along which the system can move away from the symmetry point. In pictorial terms one can think of a growing number of 'downhill' paths in the free energy surface leading away from the symmetry point. Thus a configuration $\boldsymbol{V}$ lying close to the symmetric point $\boldsymbol{V}^{sym}$ (such that $\Sigma < 1.01/N$) at a temperature far below T_c^{theo} is likely to be driven by the dynamics into the nearest free energy valley — the choice of which will be governed by the perturbation variables ε_{ia} — rather than having any chance of moving into a deep free

[6] Very stringent Δ values (e.g. $< .001$) may, however, prevent the system ever converging to a fixed point, causing premature termination of the run into a low Σ state.

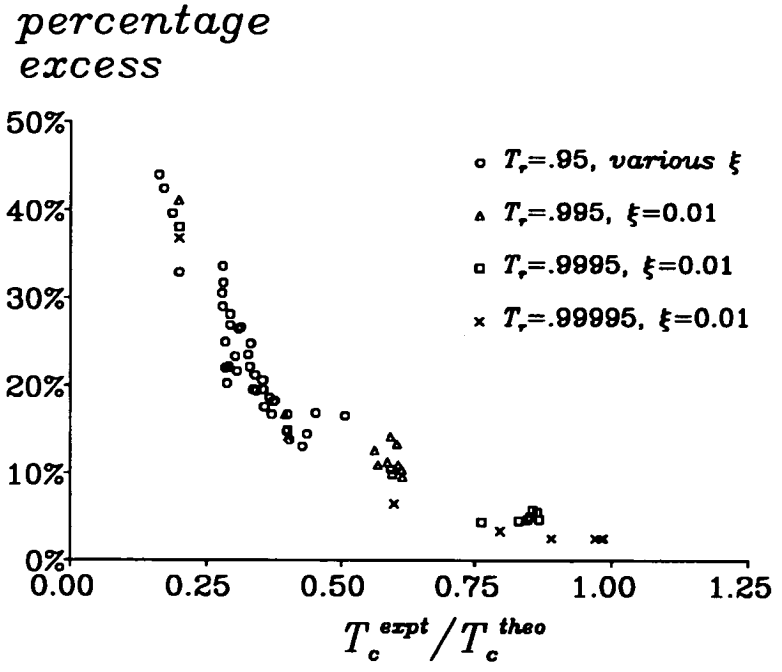


Figure 4.3: Tour-length performance plotted against T_c^{expt}. The y axis plots the mean tour percentage excess over the best Or*-opt tour for this TSP. All data points refer to $N = 50$, using the same range of T_o/T_c^{theo} values as in figures 4.1 and 4.2, with $\Delta = 1$ and $\mathcal{M} = 500$. The error bars are omitted for clarity.

energy minimum. So, runs in which T_c^{expt} is significantly below T_c^{theo} (whether through having $T_o < T_c^{theo}$ or through the secondary parameters' impact) tend to find poor free energy minima and hence generate low quality tours.

4.4.3 Experimental $T_c(\beta, \gamma)$ Behaviour

From equation 4.30, the parallel dynamics analysis predicts that $T_c(\beta)$ is a piecewise linear function, having a positive gradient for $\beta > \beta_o$ and a negative (γ dependent) gradient for $\beta < \beta_o$, where β_o is positive if $\gamma > \gamma_c$ (from equations 4.28–4.32). For the two TSP instances used in section 4.4.2, the analysis was tested by conducting runs over a range of β values for several γ settings, and measuring T_c^{expt}. Taking, as before, $\sigma_d/d = 0.478$ and $\chi = 0.65$, gives (from equation 4.32) the predicted value of γ_c as 0.27; data was collected at γ values

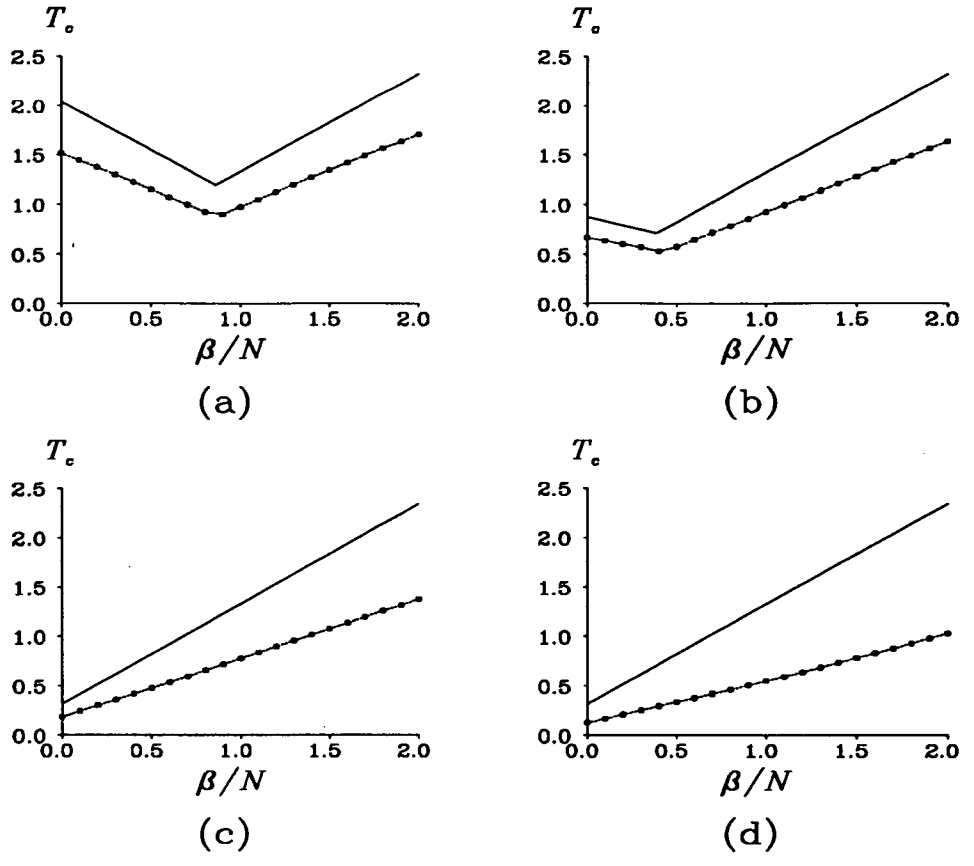


Figure 4.4: T_c plotted against β/N, for $\gamma = 1.0, 0.6, 0.2$, and 0.05, in graphs (a)–(d) respectively. Solid lines represent the theoretical profiles, dotted lines the data, the error bars are negligible and have been omitted. Parameter settings: $\alpha = 1$, $T_o = 1.2T_c^{theo}$, $T_r = .995$, $\Delta = 1$, $\xi = .01$, $\mathcal{M} = 5000$.

above and below γ_c. Figure 4.4 shows the results for $N = 50$. The $N = 200$ data overlies the $N = 50$ data almost exactly, and so has not been plotted.

Modulo the systematic scaling down from the secondary parameters discussed in section 4.4.2, the data is consistent with the theory.[7] Notice that the discrepancy between T_c^{expt} and T_c^{theo} grows as γ decreases. This trend is explicable

[7]Further runs at a few selected (β, γ) points using more stringent (Δ, T_r) settings produced T_c^{expt}/T_c^{theo} values climbing towards unity, thereby also lending support to the numerical accuracy of the analytical $T_c(\beta, \gamma)$ function.

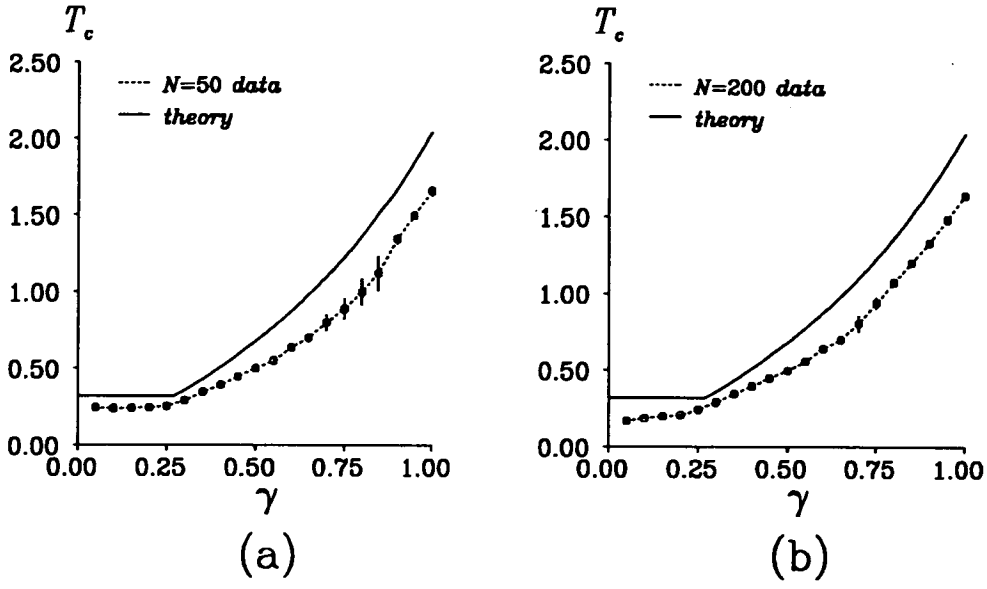


Figure 4.5: T_c plotted against γ for β set to zero. (a) $N = 50$ and $\Delta = 0.001$, (b) $N = 200$ and $\Delta = 0.01$. All other parameters are as in Figure 4.4.

through an argument similar to those given in the previous section: assuming that the system is near to $\boldsymbol{V}^{sym}$ at a temperature just below T_c, the dynamics will attempt to move the system out from that point towards a new local minimum. Initially, because the symmetric point $\boldsymbol{V}^{sym}$ has only *just* turned unstable, Δ^{expt} will be small, smaller than the Δ' threshold. Thus the implementation will only perform one sweep per T value until T has fallen some way below T_c^{theo}, rather than allowing a full relaxation at each temperature. Clearly a small step-size further restricts the system's ability to move far through phase space in these conditions, small values of γ therefore accentuate the difference between T_c^{expt} and T_c^{theo}.

In addition to giving the transition temperature, the foregoing analysis also showed that the *type* of transition should change at $\beta = \beta_o$, being oscillatory for $\beta < \beta_o$ and smooth for $\beta > \beta_o$. The numerical studies support this. For $\beta < \beta_o$, the runs terminate through lack of convergence to a fixed point at, or just below, T_c^{expt}, into states in which Σ is still only $O(N^{-1})$. For $\beta > \beta_o$, the algorithm

succeeds in generating high Σ states, though, as discussed earlier in section 4.2, if $\gamma > \gamma_c$ these states will only correspond to low quality tours, e.g. for $N = 50$ and $\gamma = 1$, the tours are about twice the optimal length.

A complementary way of assessing the $T_c(\beta, \gamma)$ theory is to vary γ whilst keeping β constant. The results of this test are shown in Figure 4.5, for $\beta = 0$. The theoretical profile (derived from equations 4.26–4.32) shows T_c falling non-linearly as γ is reduced from 1 to γ_c, and reaching a plateau below γ_c. From equations 4.29–4.31, the value of T_c on the plateau is $2\chi\sigma_d$. Comparison with experimental data is complicated by the variation in size of the scaling factor for different values of γ, as seen in Figure 4.4. Very small Δ values were employed for the Figure 4.5 runs, in order to try and minimize the reduction factor over the whole γ range. The $N = 50$ data is in reasonable agreement with the theoretical profile, particularly for low γ — the T_c curve flattens out close to the expected γ_c position, with a value also close to that predicted. In the $N = 200$ plot, the 'plateau' region is less well defined; this is probably because these runs had to be performed with $\Delta = 0.01$ rather than 0.001, due to the very long run times using $\Delta = 0.001$.

All of the experiments reported so far were for the case of parallel dynamics. Some experiments were also conducted to check the serial T_c analysis in the $\beta > 0$ range. As in the parallel case, the experimental values of T_c were found to be depressed by secondary parameter effects, but again, modulo this (γ-dependent) scaling down factor, in agreement with the analytical predictions. Furthermore, as predicted earlier, for very small γ, e.g. $\gamma \leq .05$, serial updating runs behave almost identically to those employing parallel updating, not just in regard to T_c^{expt}, but also in the final configurations and the number of sweeps required to find them.

4.5 Conclusions

This chapter has considered the behaviour of the mean field Potts TSP network under dynamics which implement an Euler integration scheme for an appropriate differential equation, with a user-defined step-size γ. The earlier iterative dynamics studied by [Peterson & Söderberg 1989] are a special case of these new dynamics. An analysis by the above authors of the network's critical temperature, i.e. the temperature at which the symmetric maximum entropy state turns unstable, was generalized to the case of the new dynamics, for both serial and parallel updating modes. Understanding the parameter dependence of this temperature is vital in choosing values for the network's free parameters, α, β, and γ. For example, the analysis revealed a critical step-size value, above which the algorithm has inappropriate scaling behaviour for large problems when using parallel dynamics.

Testing of the predictions for the critical temperature against simulation data was complicated by the simulations' sensitivity to various secondary parameters, e.g. the annealing rate, the level of initialization noise, and the convergence criterion. Systematic numerical study of these effects, which also have a strong influence on the final tour quality, led to an understanding of their causes. Armed with that knowledge, it was possible to verify the theoretical expressions both qualitatively with regard to the functional forms, and also, to a reasonable degree, quantitatively.

CHAPTER 5

Potts TSP Network: Parameter Optimization and Performance

Building on the work presented in the previous chapter, this chapter examines further issues in the use and performance of mean field Potts TSP networks. The chapter is organized as follows. Sections 5.1 and 5.2 present analytical methods for optimizing the values of the two free coefficients in the system's energy function. Data from numerical experiments supports the theoretical work. As well as being of practical benefit, these analyses also indicate that the energy function remains properly balanced in the limit of very large problems. Section 5.3 discusses the quality of the tours generated by the algorithm, both for Euclidean problems as well as for instances from the much harder class of non-Euclidean random metric problems. The simulations were performed using non-trivial problem sizes, i.e. TSPs with 50 or 200 cities. An explanation for the poor performance on random metric problems is proposed.

5.1 Choosing the Value of Coefficient α

From section 4.1, recall that the energy function for the Potts spin model is of the form: $E = E_t + E_\alpha + E_\beta$, where $E_\alpha = \frac{\alpha}{2}\sum_a(\sum_i S_{ia})^2$. The role of E_α is to encourage the system into a configuration in which every partition a is occupied by exactly one city, because only such configurations represent well defined tours. It does this through penalizing any non-uniformity in the distribution of the $\sum_i S_{ia}$ terms across the space of partitions. In the mean field Potts model, where now $E_\alpha = \frac{\alpha}{2}\sum_a(\sum_i V_{ia})^2$, E_α naturally plays a similar role. The strength of this restraint term is clearly governed by the value of the coefficient, α. In the original paper on the mean field Potts model, randomly distributed unit square Euclidean TSPs with $N = 50, 100,$ and 200 were all studied with α set to 1. The authors acknowledged that the appropriate value of α would be dependent on the intercity distance scale. However, a rationale for the precise manner in which α should depend on the $\{d_{ij}\}$ matrix was not given, nor was any non-empirical justification presented for the choice of $\alpha = 1$ in the particular case of unit square TSPs. This section provides a theoretical basis for choosing an appropriate value of α, valid for both serial and parallel updating dynamics.

The general principle guiding the choice of α is that it should be chosen as small as possible, yet large enough for the E_α term to provide an effective restraint. Excessively large values allow the network effectively to ignore the tour-length term in E, with the result that the network will produce configurations which do satisfy the problem's constraints (i.e. configurations which represent tours) but which are profligate in their tour-length cost. Conversely, if α is too low, it becomes energetically favourable for the network to settle into a configuration which does not represent a tour. The task therefore is to develop analytical understanding of conditions which bound the value of α from below, through the requirement that α be just sufficient to generate an effective E_α restraint term. Three distinct, but complementary approaches to this task are now presented,

followed by a numerical investigation.

5.1.1 The ΔF Approach

Consider the behaviour during the initial stages of the algorithm, when $T \approx T_c$ and the network is either in, or close to, the symmetric fixed point $\boldsymbol{V}^{sym}$, defined, recall, by $V_{ia} = 1/N \ \forall i, a$.

For simplicity, let us take the network to be at the symmetry point. The energy components can then be written (trivially) as:

$$E_t^{sym} = (N-1)d, \qquad E_\alpha^{sym} = \alpha N/2, \qquad E_\beta^{sym} = -\beta/2, \tag{5.1}$$

where d, recall, is the mean inter-city distance. Note here that, given the Potts constraint (4.11), E_α^{sym} is the minimal value of E_α. For sufficiently low α, clearly $E_t^{sym} \gg E_\alpha^{sym}$, so the system will place more emphasis on reducing E_t than on maintaining E_α at its floor value. As E_t only involves pairwise spin products between adjacent partitions, observe that any configuration in which the $\sum_i V_{ia}$ terms alternate in a 0-2-0-2... pattern over the partitions has $E_t = 0$, the minimum possible value of E_t. Such a pattern would, however, cause E_α to rise to αN. Consider in particular the configuration, henceforth termed the '0-2-0-2' state, which is defined by:

$$V_{ia} = \begin{cases} 2/N & \text{for } a = (2k) \bmod N, \qquad k \in Z, \quad \forall i \\ 0 & \text{for } a = (2k+1) \bmod N, \quad k \in Z, \quad \forall i \end{cases} \tag{5.2}$$

The net change in E (ignoring ΔE_β which is of a lower order) in shifting from the symmetric state to the '0-2-0-2' state would be

$$\Delta E = N(\alpha/2 - d) + d \ \approx N(\alpha/2 - d) \tag{5.3}$$

where the approximation is valid for large N. So provided that $\alpha > 2d$, the symmetric state will have a lower internal energy than the '0-2-0-2' state. However,

one must remember that self consistent solutions of the mean field equations are minima, not of the internal energy but of the free energy, F. Therefore the entropies of the two states must also be considered. Rather than using the expression identified in equation 4.10 as the entropy, it is simpler to exploit the connection between statistical mechanics and information theory [Jaynes 1957], by using the information-theoretic entropy measure, $-\sum_i^n p_i \ln p_i$. This measures the amount of uncertainty associated with a process having n possible outcomes, where p_i is the probability of outcome i.

By interpreting V_{ia} as the probability of city i being in partition a, the uncertainty about the position in the tour of city i can be written as $-\sum_a V_{ia} \ln V_{ia}$, and hence the total uncertainty about which tour is represented by the configuration written as:

$$\mathcal{S} = -\sum_i \sum_a V_{ia} \ln V_{ia} \tag{5.4}$$

This leads to $\mathcal{S}$ values of $N \ln N$ and $(N \ln N - N \ln 2)$ for the symmetric and '0-2-0-2' states respectively. So a transition from the symmetric state to the '0-2-0-2' state will increase the free energy, provided $\alpha > 2(d - T \ln 2)$. Thus this ΔF argument suggests that $2d$ is a safe choice for α. The reasons why the '0-2-0-2' state must be avoided are twofold. Firstly, because as T is lowered they do not produce valid tours — alternate partitions are left unvisited by any city. However, as indicated in section 4.4.1, it is always possible to infer a possible tour from any configuration. But even if this is done, the length of the resulting tour will still be far from optimal, because with E_t minimized to zero early on, there is nothing subsequently to drive the system towards a low cost tour.

One must, however, be cautious in drawing conclusions from this argument. Whether the network will actually move to a state of lower free energy depends on the dynamics. Recall from section 4.1.4 that we were unable to prove that F was decreased by the dynamics in the $\gamma \to 0$ limit, even though minima of F are fixed points of the dynamics. Therefore there is no *guarantee* that the network

will evolve into a nearby state of lower F, even if there is no intervening energy barrier between the two states.

5.1.2 The Self-Consistency Approach

The ΔF approach found the parameter conditions under which the unwelcome '0-2-0-2' state was energetically unfavourable. The second approach, detailed below, seeks a bound on α through asking the question: under what conditions is the '0-2-0-2' state a self-consistent solution of the mean field equations 4.9?

From the definition in (5.2), the corresponding set of U_{ia} variables, defined by (4.9), are given (for all i) by

$$U_{ia} = \begin{cases} (2\beta/N - 2\alpha)/T & \text{for } a = (2k) \bmod N, \quad k \in Z \\ -4d_i/T & \text{for } a = (2k+1) \bmod N, \quad k \in Z \end{cases} \tag{5.5}$$

where d_i denotes $\sum_j d_{ij}/N$. d_i will henceforth be approximated by d in this section. To be a self-consistent solution of the MF equations, the variables must satisfy

$$\frac{e^{U_{ia}}}{\sum_b e^{U_{ib}}} = V_{ia} \qquad \forall i, a. \tag{5.6}$$

Because of the periodicity in the '0-2-0-2' state defined in (5.2), one need only consider the validity of (5.6) for a single V_{ia} term. Picking a term with value $2/N$, and assuming that β has been chosen so as to scale at most sub-linearly with N, then for large N the left hand side can be expressed as

$$\begin{aligned} \frac{e^{U_{ia}}}{\sum_b e^{U_{ib}}} &= \frac{e^{-2\alpha/T}}{N/2\left(e^{-2\alpha/T} + e^{-4d/T}\right)} \\ &= V_{ia}\left(1 + e^{(2\alpha - 4d)/T}\right)^{-1}. \end{aligned} \tag{5.7}$$

Thus there is self-consistency, for fixed T, only if $\alpha \ll 2d$. Notice incidentally, that when $\alpha = 2d$, the above equation yields $e^{U_{ia}}/\sum_b e^{U_{ib}} = 1/N$, indicating that the '0-2-0-2' state would revert to the symmetric fixed point under $\gamma = 1$ parallel dynamics.

In summary, by choosing $\alpha \geq 2d$ the free energy landscape can be engineered so that the troublesome '0-2-0-2' state is neither a self-consistent solution of the MF equations nor is it of lower free energy than the symmetric state.

5.1.3 The T_C Mode Approach

This third method considers the consequences when one of the assumptions made in reaching the $\lambda_{\max/\min}$ estimates of (4.26), no longer holds.

Recall that $\lambda_{\max}$ is $\max_{k\neq 0}(\lambda_\perp^k)$ provided the (4.25) inequality, i.e. $\alpha > 2(d - \sigma_d\chi)$, is satisfied. If α is too low for this inequality to hold, then $\lambda_{\max}$ becomes $\lambda_\parallel^{N/2}$. In this case, assuming that β is taken larger than β_o, as T is lowered through T_c, $\hat{\varepsilon}_{N/2}$ will be the first Fourier perturbation vector to grow. It will grow parallel to the $(1,1,1,\ldots)$ eigenvector associated with the $\lambda_\parallel$ eigenvalue. Transforming back to the real ε_{ia} variables, this means that for every city i, the $\{\varepsilon_{ia}\}$ form a wave pattern over the partition space, with wavelength equal to $\Delta a = 2$. As the V_{ia} variables are subject to the Potts constraint there is a limit to the amplitude of the wave, so that the limiting state has the V_{ia} alternating in a $0,\ 2/N,\ 0,\ 2/N\ldots$ fashion. But this is just equivalent to the state defined by (5.2), i.e. the '0-2-0-2' state.

Thus, using either parallel or serial updating in the $\alpha < 2(d - \chi\sigma_d)$ region requires the previously explored T_c theory to be modified so as to take account of the different $\lambda_{\max}$ value. This section has shown that one consequence of an appropriate T_c analysis for such low α, is that the transition away from the symmetry point leads directly into the '0-2-0-2' state. Again, choosing a value of $2d$ for α should enable the system to avoid this problematical fate.

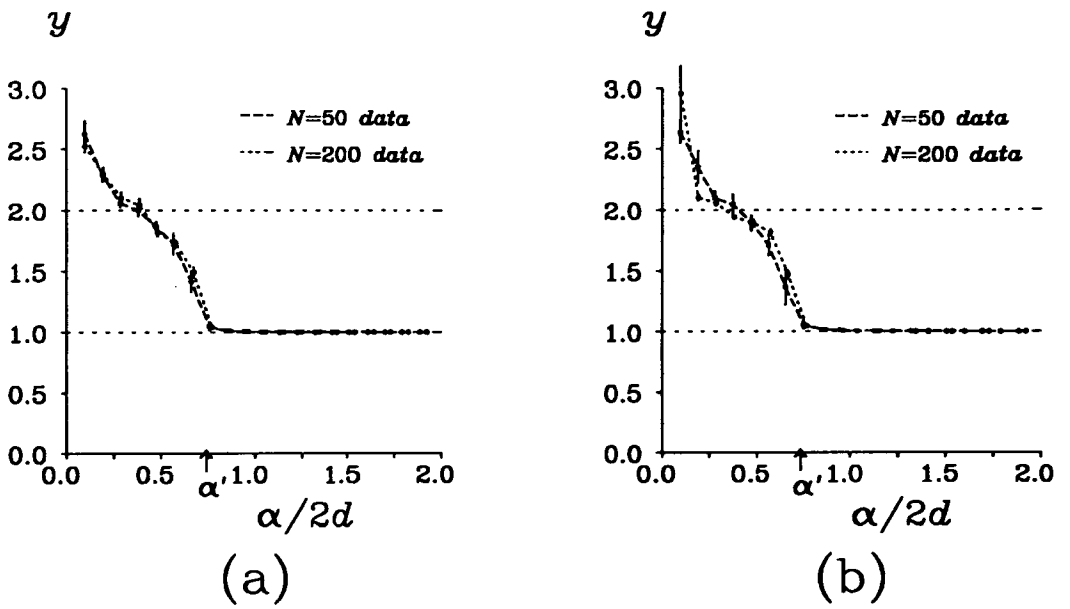


Figure 5.1: Plot of restraint energy against α. The ordinate, y, denotes $(2E_\alpha/\alpha N)$. The dotted lines highlight the y values of 1 and 2, corresponding to the minimal E_α value and the value for 0-2-0-2 type states, respectively. α' denotes the point $\alpha = 2(d - \chi\sigma_d)$, computed using the values in section 4.4.2. Parameter settings: $\beta = 1$, $\gamma = 0.1$, $T_o = 1.2T_c^{theo}$, $T_r = 0.95$, $\Delta = 1$, $\xi = 0.01$, $\mathcal{M} = 100$. (a) Serial updating case ($\gamma = 1$ gives a similar plot, data not shown). (b) Parallel updating case; the γ value is low enough to ensure that the oscillatory transition mode is avoided throughout the α range.

5.1.4 Numerical Study of Low α Behaviour

All three of the approaches just examined suggest that at some value below $2d$, α ceases to be able to generate an effective E_α restraint term. Thus $2d$ emerges as a natural theoretical choice for α. To test this, runs were performed over a range of α values, and the final values of E_α, E_t and the tour-length recorded. Figure 5.1 shows the final E_α data for both types of updating scheme. Within the flat region, E_α is at its floor value, indicating that the restraint term is being effective. Below $\alpha/2d \approx 0.8$, E_α rises sharply, indicating that α is too small to be effective. The data points with y values between one and two reflect configurations with only a partial 0-2-0-2 pattern in the $(\sum_i V_{ia})$ sums, likewise

those of value greater than two are due to a mixture of 0-2-0-2 and 0-3-0-0-3 patterns. Other measures, such as the tour-length and final value of E_t, show similarly sharp changes below $\alpha = 2d$.

Notice that the boundary between the two α regimes lies, as predicted, somewhat below $\alpha = 2d$, and close to α', the value at which the critical temperature analysis predicted a change in the behaviour. In conclusion, for optimal performance the value of α needs to be tied to the $\{d_{ij}\}$ matrix through the relationship: $\alpha \approx 2d$, where d is the mean inter-city distance.[1]

5.2 Choosing the Value of Coefficient β

As with α, it is clearly a sensible policy to set the β coefficient small, in order to emphasise the importance of the tour-length term (E_t) within the overall energy function. However, whereas with α there was a principle guiding its choice, i.e. that it should be just large enough to generate an effective E_α restraint term, with β this is not so. The reason is that the E_β term, originally introduced in the Hopfield-Tank type networks to reward configurations in which each city is located in exactly one partition (as in equation 4.2), is strictly speaking redundant in Potts type models, where this property is enforced as a constraint (by imposing $\sum_a S_{ia} = 1$ or $\sum_a V_{ia} = 1$). Consider the E_β expression for the Potts spin model (equation 4.4): $-\frac{\beta}{2}\sum_{ia} S_{ia}^2$. As the $S_{ia} \in \{0, 1\}$, any state respecting the Potts constraint will have a constant E_β value ($= -N\beta/2$). However, rather than simply discarding E_β altogether, Peterson and Söderberg retained it, and demonstrated that a particular β regime (typically $\beta \sim N$) avoids the oscillatory transition mode at T_c for $\gamma = 1$ parallel dynamics (as detailed in section 4.2). Unfortunately, although in the Potts spin model E_β is constant, in the mean

[1]This accounts for the suitability of Peterson and Söderberg's choice of $\alpha = 1$, since the uniformly random unit square TSPs they studied have $d = 0.521$ in the $N \to \infty$ limit.

field Potts model, where $S_{ia} \mapsto V_{ia} \in [0,1]$ and $E_\beta = -\frac{\beta}{2}\sum_{ia} V_{ia}^2$, E_β is no longer constant — it is related to the saturation through $E_\beta = -N\beta\Sigma/2$. So during a complete run, ΔE_β is $O(N\beta)$, compared to a ΔE_t of typically $O(N)$. Thus, using β in the above manner is impractical: although it does select the better T_c mode, the energy function is dominated by E_β, which results in very poor tour-length performance. A new, rational, approach to the issue of avoiding the oscillatory T_c mode with parallel dynamics, by selection of the step-size γ rather than β, was presented in section 4.2.

So E_β, in both the spin and mean field versions of the Potts model, is redundant, in the sense that the property it was originally designed to enforce is now contained as an explicit constraint. But whereas in the Potts spin model E_β also remains constant, in the mean field model it can vary. Thus the value of β does matter in this case, because it affects the balance of terms in the energy function. Ideally therefore, we would like to set β to zero, reducing E to just two terms: the tour-length cost term (E_t) and a genuine restraint term (E_α). The oscillatory transition mode could still be avoided, by setting $\gamma \leq \gamma_c$ for parallel dynamics (for serial dynamics this mode is irrelevant). So should β just be set to zero? In their work, Peterson and Söderberg kept β at 0.5 (with serial dynamics) without justifying this choice, though they did state that the presence of β had a "constructive balancing effect in solving the MFT equations" [Peterson & Söderberg 1989, page 14]. Further numerical work to be presented later broadly supports this choice: if β is chosen any smaller, or zero, the system is unable to evolve into (high Σ) states representing well defined tours. The analysis in the following sections, based on an examination of self-consistency in the MF equations, accounts for this behaviour, in both qualitative and semi-quantitative terms.

5.2.1 Basic Self-Consistency Argument for β

Consider a configuration which represents a perfectly defined tour, i.e. one in which $\forall i, \exists a_i : V_{ia} = \delta_{aa_i}$ with $a_i = a_j$ iff $i = j$. Our goal is to discover the parameter conditions under which this tour state constitutes a solution of the MF equations 4.9.

The argument is valid for both Euclidean and non-Euclidean $\{d_{ij}\}$ distributions. It does however make two assumptions. Firstly, it will only consider self-consistency within the set of mean field variables associated with a single city. Therefore the validity of the results will depend on the extent to which this single city is 'typical' of the cities in the problem. For simple city distributions this is not a serious problem, but for TSPs containing irregular clusters or outlying cities this caveat must be borne in mind. Secondly, it is implicitly understood that α has been chosen sufficiently large such that E_α is at its minimal value.

For notational convenience let the cities be relabelled such that the desired tour is represented by the above configuration, with $a_i = i \ \forall i$. From equation 4.9, the MF components for city i are completely self-consistent if $\exp(U_{ii}) = \sum_b \exp(U_{ib})$. It is more useful however, to consider complete self-consistency as the limit: $\rho \to \infty$, where ρ is defined by

$$\rho = \frac{e^{U_{ii}}}{\sum_{b \neq i} e^{U_{ib}}} \tag{5.8}$$

Thus ρ should be regarded as gauging the degree of self-consistency. So to achieve a particular value of ρ, the following equation must be true (using equation 4.9):

$$e^{(-2d'+\beta)/T} = \rho \sum_{j \neq i} e^{(-d_{i,j-1}-d_{i,j+1})/T} \tag{5.9}$$

where the bondlengths within the tour, i.e. the distances between adjacent cities in the tour, have been modelled by a single term, d'. Denoting the $\sum_{j \neq i}$ sum by $\mathcal{G}$, it is clear that for large N, $\mathcal{G}$ is bounded as follows: $Ne^{-2D'/T} < \mathcal{G} < N$, where

$D' \equiv \max_{i,j}(d_{i,j})$. Combining this with the above equation leads to the following conditions on β in order achieve a particular ρ for this tour:

$$T \ln N\rho - 2D' + 2d' < \beta < T \ln N\rho + 2d' \tag{5.10}$$

This result, though derived from a simple argument, contains several important features.

- For any particular tour state, the degree (ρ) to which that state is self-consistent is related (exponentially) to β. This is consistent with the observation from numerical studies that the system cannot evolve into states resembling tours if β is too low.

- The quality of the tour enters (5.10) through the term $2d'$ since, recall, the tour-length is being modelled by Nd'. Thus, for fixed β, (5.10) implies that short tours are more self-consistent than longer ones, and therefore more liable to be found by the algorithm. This is a very useful property: rather than relying solely on the E_t term to select out low cost tours, the presence of β also contributes to this process, by tending to prevent high cost tours from being self-consistent solutions of the mean field equations. A similar positive role for the β coefficient was previously found for a related mean field Potts model [Gislén *et al.* 1989] designed for scheduling problems, though the justification was not given in terms of a self-consistency argument, as here.

- β needs to scale as $\ln N^T$. This is a potentially worrying trend, since it leads to the $\Delta E_\beta / \Delta E_t$ ratio being of $O(\ln N^T)$, implying a diminishing importance for the tour-length term in the overall energy function for large N. Replacing T by T_c gives one upper bound on the trend (section 5.2.4 discusses another bound). For the classes of city distributions discussed in section 4.2, using parallel dynamics with $\gamma < \gamma_c$ or serial dynamics, T_c is given by $2\chi\sigma_d + (\beta - 2d)/N$. Using this in the right hand inequality of

(5.10), and retaining only the leading term in β, leads to an absolute upper bound on β of $2\chi\sigma_d \ln N\rho + 2d'$. For the uniformly random unit square TSPs used as testbeds in this study, $2\chi\sigma_d \approx 0.32$, indicating that even in this worst case, the growth of β with N would be quite modest. In practice, configurations representing tours only develop after the temperature has been annealed some way below T_c. Empirically, for the testbed TSPs, high Σ tour states tend to freeze out at $T \approx 0.1$, independent of N; this suggests that, at least for this class of TSP, the growth in β with N may not be practically significant.

This simple analysis is clearly of value in identifying the key qualitative effects of the parameter β and how it needs to scale in order for tours to be solutions of the MF equations. However, because of the relatively crude nature of the $\mathcal{G}$ bounds, (5.10) only manages to pin β down to within a window of width $2D'$. As D' is the maximum inter-city spacing, the analysis evidently has only limited value in suggesting what value β should be given in order for good tours to be possible solutions. A more refined method for estimating $\mathcal{G}$, and hence selecting β, is now presented.

5.2.2 Refined Self-Consistency Argument for β

This argument follows the previous one up to equation 5.9. The problem is that exact evaluation of $\mathcal{G}$ requires foreknowledge of the city ordering in a particular tour. For large TSPs with distances obeying the triangle inequality (e.g. Euclidean TSPs) this problem can be avoided, as follows. First, note that the triangle inequality gives[2]

$$d_{i,j+1} + d_{j,j+1} \geq d_{ij} \geq d_{i,j+1} - d_{j,j+1} \tag{5.11}$$

[2]Recalling that this notation labels the cities in their order on the tour.

and a similar condition for $d_{i,j-1}$. Restricting ourselves to near-optimal tours, in which the mean bondlength (d') within the tour is generally of a lower order than the mean inter-city spacing (d), these inequalities imply that $d_{i,j\pm1} \approx d_{ij}$ for a typical city i. Thus $\mathcal{G}$ can be approximated by $\sum_{j\neq i} \exp(-2d_{ij}/T)$, which does not include any dependence on a particular tour. $\mathcal{G}$ can then be estimated numerically for any particular TSP (given a sensible estimate for T). If, however, the cities have been drawn from — or can be reasonably modelled by — a distribution function, or, in the case of non-geometrical TSPs which still obey the triangle inequality, if there is a distribution function for the d_{ij} terms, then the above approximation to $\mathcal{G}$ can be evaluated as an integral in the limit of $N \to \infty$. This continuum result can then be taken as an estimate for finite N problems. The validity of this approach will clearly depend on N being large and the distribution function being accurate. (For TSPs which do not obey the triangle inequality the method just outlined is not applicable; Appendix B includes a different approach for one class of TSPs in this category). This section illustrates the integral approach, for the case of Euclidean TSPs with cities drawn from a uniformly random distribution in a square of side 1.

Letting $\boldsymbol{w}$ denote the location of city i, $\boldsymbol{z} \equiv (x, y)$ that of city j, and $d(\boldsymbol{w}, \boldsymbol{z})$ the distance between them, $\mathcal{G}$ is given by $N\mathcal{I}$, where

$$\mathcal{I} = \int_0^1 \int_0^1 \exp[-2d(\boldsymbol{w}, \boldsymbol{z})/T] \mathrm{d}x \mathrm{d}y. \tag{5.12}$$

By choosing $\boldsymbol{w}$ to lie at the centre of the square and exploiting the ensuing symmetry, $\mathcal{I}$ can be expressed in polar form as

$$\mathcal{I} = 8 \int_0^{\frac{\pi}{4}} \mathrm{d}\theta \int_0^{\frac{\sec\theta}{2}} r e^{-2r/T} \mathrm{d}r. \tag{5.13}$$

After integrating by parts and making the substitution $x = \sec\theta$, $\mathcal{I}$ is given by $(\pi T^2/2 - C)$, where

$$C \equiv 2T^2 \int_1^{\sqrt{2}} \frac{e^{-x/T}}{x\sqrt{x^2-1}} \mathrm{d}x + 2T \int_1^{\sqrt{2}} \frac{e^{-x/T}}{\sqrt{x^2-1}} \mathrm{d}x. \tag{5.14}$$

	T		
N	0.1	0.2	0.3
50	0.42	0.90	1.49
100	0.43	0.98	1.64
500	0.50	1.22	2.04
1000	0.55	1.34	2.22

Table 5.1: Values of β^* for various N and T, computed for $\rho = 10$, taking a value of $0.75/\sqrt{N}$ for d' (following the discussion in section 1.6).

Although these integrals cannot be computed analytically, they are obviously positive and can be shown to be bounded above by:

$$\mathcal{C}_{U.B.} = 2Te^{-1/T}\left(\ln(\sqrt{2}+1) + \pi T/4\right). \tag{5.15}$$

In the appropriate temperature range[3] i.e. $T < T_c(= .32)$, $\pi T^2/2$ exceeds $\mathcal{C}_{U.B.}$ — only by a factor of five at T_c, but by over an order of magnitude in the more relevant $T \approx 0.1$–0.2 range. Thus the $\mathcal{C}$ term can be dropped, giving $\mathcal{G} = N\pi T^2/2$ and hence the final expression for β^* — the value of β which will allow near-optimal tour states to have a self-consistency ρ — using (5.9), as

$$\beta^* = T\left(\ln N\rho + \ln\frac{\pi T^2}{2}\right) + 2d'. \tag{5.16}$$

Table 5.1 lists some values for β^*. Observe that for low T the rise in β^* with N is very small, and secondly that the values at low T are close to 0.5, Peterson and Söderberg's choice for β.

There are several minor problems in using equation 5.16 directly to select β:

- It was derived from a consideration of the self-consistency within the MF components for a particular city, located at the centre of the distribution.

[3]Approximating T_c in the low β regime by $T_c(\beta = 0)$, assuming either serial dynamics, or parallel dynamics with $\gamma < \gamma_c$.

Repeating the derivation for the other extremal case, i.e. a city at a box corner, gives a similar result but with a $\ln \frac{\pi}{8}$ term, rather than $\ln \frac{\pi}{2}$. β should therefore be chosen between these two expressions for β^*.

- What is an appropriate value for ρ? By definition, absolute self-consistency only holds in the $\rho \to \infty$ limit. In practice however, what matters is whether tour states are self-consistent *enough* to be found by the algorithm; thus ρ values of say 10 or 100 might suffice. Fortunately, as ρ enters (5.16) logarithmically, β^* is fairly insensitive to the precise value of ρ.

- The analysis was based on applying a self-consistency criterion to clean ($\Sigma = 1$) tour states. In practice though, the algorithm is terminated before the saturation becomes one, because the tour structure is essentially fixed by the time Σ reaches about 0.8 or 0.9. Ideally therefore, the analysis should be performed on states having such values of Σ. However, the calculations are messier in that case, so the original analysis will be used here as an approximation to it.

These points lead to some uncertainty in setting the absolute value of β but do not alter the scaling behaviour with N. As was implied both in section 4.2 and at the start of the current section, for the algorithm to be properly energetically scalable the value of β ought to be $O(1)$. Therefore the analytical $\ln N^T$ scaling appears problematical. However, the low empirical final temperatures suggest that this scaling will be fairly benign, a view supported by the numerical experiments reported next, and underpinned by further work on the final temperature presented in section 5.2.4.

5.2.3 Numerical Study of Low β Behaviour

To test these predictions, runs were performed over a range of low β values, and the final saturation (Σ) values recorded. As usual, the TSP instances were

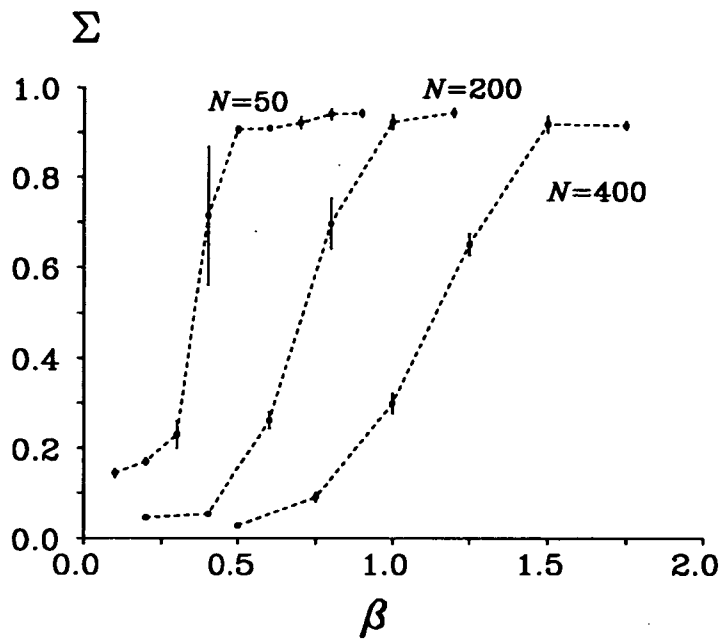


Figure 5.2: Final saturation value Σ plotted against β. The fact that the curves flatten out above 0.9 rather than at 1.0 is an artefact due to the termination criteria used (see section 4.4.1 for details). Due to the long run times, each $N = 400$ data point represents the average over five trials rather than the usual ten. Parameter settings: $\alpha = 1, \gamma = 0.2, T_o = 1.2T_c^{theo}, T_r = 0.95, \Delta = 1, \xi = 0.01$, $\mathcal{M} = 200$, parallel updating.

---

from unit square random distributions, with $N = 50, 200, 400$ (one instance of each).

Figure 5.2 shows some of the data. Observe in the 50 city case the sharp transition between values of β which are able to generate high saturation states, and those (lower) values which are not. This is consistent with the prediction that the self-consistency of a tour state (and hence its likelihood of being found) should vary rapidly with β. For $N = 200$ and 400, the transition is not so sharp, and occurs at an increasingly large β value, contrary to the predictions. This is, however, an artefact due to an implementation decision (see section 4.4.1) to terminate runs which fail to converge within $\mathcal{M}$ sweeps at any particular temperature. Upon inspection, runs terminating into intermediate (i.e. $\approx 0.2 - 0.8$) Σ states were found still to be evolving — albeit slowly — towards high Σ

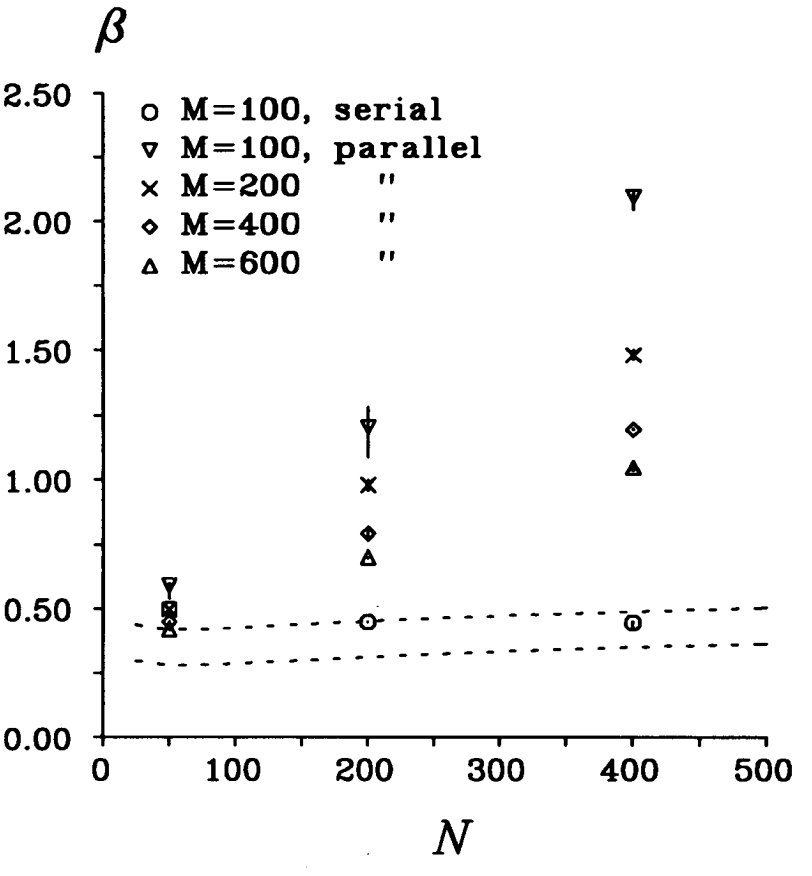


Figure 5.3: Comparison between the experimental and theoretical low β results. The data points represent the experimental β' values and their associated uncertainties. Except for $\mathcal{M}$, the parameter values are the same as in Figure 5.2 (serial updating runs used $\gamma = 1$). β^* values (using $\rho = 10, T = 0.1$) for cities at the centre and corner of the box are shown by the upper and lower dotted lines respectively.

configurations at the time of their premature termination. Increasing the $\mathcal{M}$ cutoff threshold both sharpens up the transition and brings it down to a lower β range, as demonstrated by Figure 5.3. This figure plots the minimum value of β (denoted β') found capable of generating final states with mean $\Sigma = 0.9$,[4] against N, for several $\mathcal{M}$ values. Also shown are β^* theoretical trend lines. The parallel updating runs using low $\mathcal{M}$ values deviate markedly from the β^* predictions. But this is not a valid test of the β^* analysis, since the experimental results are being influenced by an artificial dynamical cutoff, whereas β^* was derived from a purely static analysis. As the cutoff becomes less important (i.e. as $\mathcal{M}$ increases), so the experimental trend moves towards the β^* range, suggesting that the two might indeed coincide in the large $\mathcal{M}$ limit. Support for this comes from the runs performed using serial updating; these give β' values consistent with the β^* predictions.

The difference here between runs using serial rather than parallel updating is largely due to the necessity of using small step-sizes in the latter scheme. As explained in section 4.4.3, low γ runs typically only move away from the symmetry point at temperatures substantially below T_c^{theo}, by which time the free energy minima are located in medium-high Σ regions. Thus, to reach one of these minima the system has to move a large distance through phase space; this, coupled with the small step-size, leads to the necessity of large $\mathcal{M}$ values for parallel updating runs to succeed.

As well as supporting the self-consistency predictions for β, numerical experiments also confirm the complementary suggestion, that for optimal performance β ought to be chosen no larger than is necessary to satisfy the self-consistency requirement. Increasing β from 0.5 to only 1.5 causes a marked deterioration in the quality of the tours found, e.g. for 50 cities the average tour percentage excess rises from 7% to 23% ($T_r = .995$, serial $\gamma = 1$ updating).

[4]Calculated using a simple linear interpolation scheme on plots of Σ vs. β, such as Figure 5.2.

Thus the issue of choosing appropriate values of β appears to be rather more complex than suggested in PS. Numerically, there is definitely a fairly narrow range of β values which gives 'optimal' performance, in the sense that values below this range cannot generate tour states whereas values above this range generate increasingly long tours. The analytical work in the previous two sections accounted for the existence of this optimal β regime and also located its value reasonably accurately.

5.2.4 Dependence of β on T

The final temperature, T_f, is clearly relevant to any discussion of suitable values of β, for two reasons. Firstly, using the methods developed in sections 5.2.1 and 5.2.2 for setting β requires an estimate of T_f. Secondly, on the theoretical side, information about the expected values of T_f is vital to understanding whether the required $\ln N^T$ scaling of β will be harmful to the algorithm's performance on very large problems.

Until now, we have either used T_c as a crude upper bound on T_f, or taken empirical values for T_f (as in the previous section); clearly these approaches are somewhat unsatisfactory. Study of the free energy landscape reveals another characteristic temperature, T^*, which, as well as defusing the $\ln N^T$ scaling issue and giving an improved T_f upper bound for many types of TSP, also helps in understanding why the mean field Potts algorithm performs very poorly on certain types of TSP.

Consider the relative values of the free energy at the maximum entropy symmetric state $\boldsymbol{V}^{sym}$ and at a hypercube vertex representing a tour of length $\mathcal{L}$. Using equations 4.6 and 4.7, it is straightforward to deduce that the tour vertex

has the lower free energy for T satisfying

$$T < \frac{d + \beta/2 - \mathcal{L}/N}{\ln N} \tag{5.17}$$

The highest such temperature, denoted T^*, occurs for those vertices representing the optimal tour.[5] Assuming that the d_{ij} terms have been scaled to be $O(1)$, $\mathcal{L}_{opt}$ is certainly $O(N)$, thus T^* is $O(1/\ln N)$. Two cases now arise:

$T^* < T_c$ In this case T^* can be used to give an improved upper bound on T_f. To see this, consider how the system evolves as T is annealed. As T is brought below T_c, $\boldsymbol{V}^{sym}$ turns unstable and the system flows towards a new minimum with lower free energy and entropy; during subsequent annealing the system tracks this minimum as it shifts towards the hypercube vertex. Assuming that both E and $\mathcal{S}$ are non-increasing quantities over this trajectory (a reasonable conjecture supported by experiments), it can be shown that throughout the trajectory the system's free energy is less than or equal to that at the state $\boldsymbol{V}^{sym}$. Thus, tour vertices only become accessible at temperatures below T^*, hence this is an upper bound on T_f. (The types of Euclidean TSPs considered in section 4.2 fall into this category).

$T^* > T_c$ In this case T_c — which must also be $O(1/\ln N)$ — gives the better bound on T_f. (The non-Euclidean random metric TSPs discussed in Appendix B lie in this category).

The key point here, is that in both cases T_f has been shown to be $O(1/\ln N)$, thereby revealing the $\ln N^T$ component of β to be a harmless $O(1)$ term. This is an important result, as it implies that the terms in the energy function (4.7) remain well balanced in the thermodynamic limit.

[5] Recall that there is a $2N$-fold degeneracy, due to redundancy in the problem representation.

5.3 Performance

This section discusses the quality of the solutions found by the algorithm, how the quality depends on the annealing schedule and possible modifications to enhance the performance. In the original PS paper, the mean field Potts algorithm was tested on uniformly random Euclidean problems ($N = 50, 100, 200$) and found to generate tours which were on average only 8% longer than average simulated annealing tours. However, in a subsequent benchmark study [Peterson 1990] the Potts method was pitted against, amongst others, a sophisticated genetic algorithm. In this case, the differences in performance were 19%, 16% and 21% for $N = 50$, 100 and 200 respectively; indicating that the early optimism was perhaps due to a poor simulated annealing implementation. One of the motivations behind the current study was to clarify the issue of the algorithm's potential and actual performance.

Two very different types of TSP were studied: the 'classic' Euclidean TSPs used throughout this chapter, and non-Euclidean random TSPs. For both types of problem, the optimal tour-length was estimated through the same procedure as used for the elastic net performance study in section 3.4, i.e. the shortest tour found over fifty runs of the Or*-opt algorithm was deemed 'optimal'. Clearly this typically overestimates the optimal length, particularly for large N, but it still gives a useful basis for comparison.

With a computational complexity of $O(N^3)$ per *sweep*, software implementations of the mean field Potts algorithm are unlikely to be competitive with conventional serial algorithms such as 3-opt and Lin-Kernighan, which empirically have *overall* $O(N^2)$ running times [Johnson 1990]. This would change however if the Potts algorithm could be implemented in analogue hardware. Ideally therefore one would like to study the likely hardware performance through numerical simulations using a small step-size γ to approximate continuous time

dynamics. However, this is computationally very expensive: not only do the simulation times grow as $1/\gamma$, but there is also the problem mentioned in section 4.4.3, that low γ runs tend to depress T_c^{expt} artificially and thereby degrade solution quality. The latter point can be compensated for by more stringent annealing schedule parameters, but only at the cost of longer simulation times. Because of these problems, no attempt has been made to check the solution quality for low γ. All of the data presented in the following sections was obtained using $\gamma = 1$ with serial updating.

5.3.1 Uniformly Random Euclidean Metric TSPs

The six examples each for $N = 50$ and $N = 200$ studied in section 3.4 for the elastic net were again used here. Tables 5.2 and 5.3 show the average percentages by which the solutions found by the algorithm exceed the 'optimal' tour-length, for a range of annealing schedules. For both values of N the solution quality improves substantially with slower cooling, though of course this entails an increased number of sweeps. Counting only the sweeps after T_c^{expt}, this number grows from typically about 60–80 for $T_r = 0.95$ to approximately 250 and 2500 for $T_r = 0.995$ and 0.9995 respectively. Whilst the quality of the $N = 50$ tours is quite acceptable (being only slightly worse than the TLA-optimized elastic net tours), the same is not true for the $N = 200$ sets. Why the solution quality degrades as the problem size increases, remains an open question. Even slower cooling would very likely further improve the quality. However, the running times on these, and larger instances, would become prohibitive for software implementations. Furthermore, it remains unclear to what extent very slow annealing can bring the system close to finding *the* optimal tour. Whereas simulated annealing, given certain conditions, guarantees convergence to the optimal solution for asymptotically slow annealing [van Laarhoven & Aarts 1987, chapter 3], such a result is not true for mean field annealing algorithms, because

$N = 50$	Annealing rate, T_r		
Sets	0.95	0.995	0.9995
i	9.2 ± 5.1	7.1 ± 3.6	2.5 ± 0.0
ii	11.9 ± 2.5	8.0 ± 2.0	5.7 ± 1.7
iii	8.3 ± 5.0	4.6 ± 1.8	$2.5 \pm .07$
iv	9.0 ± 3.0	5.7 ± 1.4	5.5 ± 1.1
v	11.8 ± 3.6	11.5 ± 0.5	7.1 ± 1.0
vi	8.1 ± 6.3	5.0 ± 4.8	$0.7 \pm .07$
Mean	9.7%	7.0%	4.0%

Table 5.2: Tour-length percentage excess data for $N = 50$. Entries denote the mean and standard deviation values over ten trials for each set. Parameter settings: $\alpha = 1, \beta = 0.5, \gamma = 1, T_o = 1.2T_c^{theo}, \Delta = 1, \xi = 0.01, \mathcal{M} = 100$.

$N = 200$	Annealing rate, T_r		
Sets	0.95	0.995	0.9995
i	30.1 ± 4.2	21.6 ± 2.5	16.3 ± 1.2
ii	32.1 ± 3.8	21.7 ± 3.0	15.3 ± 2.3
iii	36.4 ± 4.8	24.8 ± 3.4	16.9 ± 1.6
iv	30.2 ± 3.2	22.7 ± 2.6	12.9 ± 1.7
v	30.6 ± 4.4	20.0 ± 3.4	14.0 ± 1.8
vi	33.8 ± 8.1	22.0 ± 2.0	16.0 ± 1.2
Mean	32.2%	22.1%	15.2%

Table 5.3: Tour-length percentage excess data for $N = 200$. Other details are as in Table 5.2, but with only five trials per set for $T_r = 0.9995$.

the mean field method is an approximation.

Various modifications to the algorithm were experimented with in order to try

and enhance performance. For example, the α and β parameters were allowed to adapt during a run, the rationale guiding the choice of adaptation rules being that these parameters should be as low as possible, whilst still respecting the various conditions on them, as discussed in sections 5.1 and 5.2. Another approach was to adapt dynamically the annealing schedule parameters, T_r and Δ, with the aim of allowing rapid cooling through uninteresting sections of the free energy landscape but slower cooling in critical sections, such as around the critical temperature. Whilst these approaches were found to produce shorter tours, this was usually at the cost of increased running times. Comparison between these modified methods and the original vanilla variety, when matched for run times, typically showed little or no net improvement.

5.3.2 Non-Euclidean Random TSPs

These TSPs are generated by drawing each d_{ij} element independently from a random $[0, 1]$ distribution, but retaining symmetry and zero diagonal elements. The appropriate T_c, α and β analyses for this class of TSPs are summarized in Appendix B. The optimal tours are known to be of length 2.014 in the asymptotic N limit [Krauth & Mézard 1989]; the best Or*-opt tours for the problem sizes studied here are typically within 15% of this. Table 5.4 shows the raw tour-lengths obtained by the mean field Potts algorithm using suitable parameters, for a single annealing schedule. Even for small N the average tours are poor, and for large N they typically exceed the optimal tours by a factor of at least four. The key to understanding why the performance is so poor is the relationship between T_c and T^* for this class.

As shown in Appendix B, T_c^{theo} scales as $N^{-1/2}$, so that for large N, by the time the symmetry point turns unstable, there are already hypercube vertices lying lower in the free energy landscape. It is therefore reasonable to conjecture that, at T_c for large N, the dominant minima will be located increasingly far away from

Sets	$N = 50$	$N = 100$	$N = 200$
i	3.52 ± 0.2	5.51 ± 0.3	10.3 ± 0.8
ii	3.77 ± 0.2	5.35 ± 0.5	10.6 ± 1.3
iii	3.57 ± 0.4	5.60 ± 0.2	10.3 ± 0.6
iv	3.23 ± 0.3	5.18 ± 0.2	9.2 ± 0.7
v	3.46 ± 0.3	5.53 ± 0.4	11.2 ± 1.9

Table 5.4: Tour-length data for the random metric case. Entries denote the mean and standard deviation values of the tour-lengths found over ten trials for each $N = 50$or 100 instance, five trials for each $N = 200$ instance. The optimal tours are approximately of length 2; see main text for details. Parameter settings: $\alpha = 1.1, \beta = 0.4, \gamma = 1, T_o = 1.2T_c^{theo}, T_r = 0.9995, \Delta = 1, \xi = 0.01, \mathcal{M} = 100$.

the symmetry point, towards the vertices, i.e. at states with $\Sigma \to 1$. This differs from the Euclidean case, where the minima appear to move away smoothly from the symmetry point as T is lowered through T_c. For these non-Euclidean random metric TSPs, or more generally any TSP for which $T^* > T_c$,[6] it is being suggested that the system is unable to track a minimum smoothly as it moves out to a hypercube vertex, because there are discontinuities in the dominant fixed point values of the free energy and Σ at T_c. Inspection of numerical runs provides strong support for this idea: after leaving the symmetric state at T_c, the system does not find another fixed point until Σ has risen substantially towards one, increasingly so for large N.

This discontinuity in Σ values of solutions of the mean field equations at T_c is similar in certain respects to a first-order phase transition, just as the continuous change in Σ at T_c for Euclidean problems is similar to a second-order transition (see e.g. [Mouritsen 1984] for a discussion of phase transitions). The fact that T_c is associated with a phase transition was realised by Peterson and Söderberg;

[6]For large instances with well behaved d and σ_d, this inequality is essentially equivalent to the condition that the ζ factor scales *slower* than $O(\sqrt{N}/\ln N)$, by equations 4.26, 4.27 and 5.17.

the contribution of the current work is the identification of different *types* of transition, how the type depends — through T^*/T_c — on the class of TSP, and why first order-like transitions should lead to poor performance. Finally, whilst the analogy is certainly useful, there is at least one respect in which the behaviour exhibited by the current system is significantly different from phase transition behaviour, as it is usually defined. Whereas in genuine first-order transitions T_c denotes the temperature at which the global minimum switches to a different solution branch, in our system it denotes the T at which the symmetric state turns unstable, this being an important temperature because *by design* the system is initially placed in that state. Hence if T^* lies above T_c, then between these two temperatures the algorithm is de facto forcing the system into a metastable state, something which of course would not happen to a physical system in thermal equilibrium.

It would be interesting to know to what extent the difficulties with the non-Euclidean problems are due to the inherent configuration space of tours, and how much is due to the neural network formulation. To the author's knowledge there have been no other reliable studies of Potts, or HT networks being applied to the case of non-Euclidean TSPs, with which to compare the observations made here. Although [Xu & Tsai 1991] did report data on these TSPs, obtained using the bond variable representation (as in section 1.4) mapped onto an analogue Hopfield network, there are several questionable aspects to their study. Firstly, they report that the LK algorithm finds tours typically at least 50% worse than their network, and secondly, that LK is also beaten by 2-Opt. Given that [Johnson 1990] found LK's percentage excess on TSPs of this sort, with $N = 10^4$, to be only 5.8%, these results are very surprising. Furthermore, Xu and Tsai claim that their excellent results are due to their algorithm being a neural implementation of the typically near-optimal 'assignment and patching' algorithm; but this claim is specious, since assignment and patching, though indeed an excellent algorithm, is only so for *asymmetric* TSPs [Lawler *et al.* 1985,

chapter 6].

Future work on these non-Euclidean TSPs could attempt to confirm (either empirically, or, ideally, analytically) that the dominant free energy minima at and above T_c, do indeed lie at increasing Σ for increasing N. If so, then that would suggest that the symmetry point is an unsuitable initial state. Alternatively, it may be that, in light of the section 4.1.4 discussion, this is one occasion in which the lack of a Lyapunov function for the dynamics is troublesome, since the system has to move a long way through phase space at a single temperature. Different network dynamics may therefore be worth exploring.

5.4 Conclusions

This chapter first examined the issue of how to set optimal values for the two weighting coefficients in the energy function. For the coefficient α, three distinct analyses led to the common conclusion that its value should be set to twice the mean inter-city distance. Setting the value of the second coefficient, β, turns out to be slightly more complex. An initial analysis suggested that it needs to increase logarithmically with the number of cities, a trend which would have negative implications for the algorithm's scalability. However, further analysis revealed that the $\ln N$ term had a counter-balancing$(\ln N)^{-1}$ coefficient. For both α and β, numerical simulation data obtained using non-trivial problem sizes, i.e. 50 and 200 cities, was in reasonable accord with the theoretical predictions regarding their optimal settings.

Secondly, the algorithm's performance on Euclidean and non-Euclidean TSPs was assessed. For the former class, average tour-lengths are within 10% of those of the best known tours for $N = 50$ but in the range 15% to 32% for $N = 200$ (15% being for the slowest annealing schedule). Why the performance deteriorates

for increasing system size remains unclear. For the random non-Euclidean TSPs, a well known difficult case, the tours were highly sub-optimal for all N values probed. An explanation for this poor result, based on arguments about the structure of the free energy landscape, was proposed.

CHAPTER 6

Conclusions

This thesis has explored several issues in the task of optimizing the performance of neural network algorithms designed to find good solutions to hard optimization problems. Much of the previous research on such networks has paid insufficient attention to the delicate matter of choosing suitable parameter values. That deficit, coupled with a lack of simulations on large scale problems, has led to uncertainty both about the actual performance of such networks on medium-sized problems, and also over their ability in principle to cope with the very large problems that arise in real applications.

The current study has contributed to the clarification of these issues in the context of the elastic net and mean field Potts algorithms for the travelling salesman problem. Central to both of these methods is the concept of a computational energy function comprising of two or more conflicting terms. Analyses showing how the relative and absolute weights of the competing terms influence the final network performance have been presented. The network dynamics, which specify how the system attempts to minimize the energy, also have a strong influence on the performance. New forms of dynamics have been successfully developed and analysed for both algorithms.

The analytical work on the elastic net has given a clear explanation of how the method may converge into invalid solutions for certain parameter ranges. This led to an accurate prescription for the value of the ratio of energy term coefficients most likely to give valid solutions. By developing self-adaptive descent dynamics, the quality of solutions obtainable within a fixed amount of computer time was dramatically improved. The new dynamical technique, of local step-size adaptation occurring on two levels, should also be widely applicable, for minimization of energy functions that change slowly over time. Regarding scalability, only a slight deterioration in the quality of tours occurred as the number of cities was increased from fifty to two hundred, with the tours being within 5% of those obtained using an efficient local optimization heuristic.

For the mean field Potts network, the dynamics specified by the original authors were generalized, as was their eigenvalue analysis of the network's critical temperature under those dynamics. The new temperature analysis given here reveals many relationships between the network's key parameters which help in assigning them suitable values. More detailed analyses were presented for deriving the optimal values of the two coefficients in the energy function. The methods developed may also be of use for parameter setting in other mean field networks. Regarding scalability, there is a marked deterioration in the tours for increasing numbers of cities in Euclidean problems, despite the energy function remaining well balanced in the thermodynamic limit. Why this should be so was not resolved. The failure of the algorithm on non-Euclidean problems has, however, been accounted for. In summary, the mean field Potts algorithm for the TSP is a well defined model on which a considerable degree of analytical work concerning parameter issues can be undertaken. Whether the method is truly scalable to very large Euclidean problems remains an open issue, and one which, given that the method' s computational complexity is $O(N^3)$ per update, will probably best be resolved analytically via statistical mechanics.

APPENDIX A

Parallel Implementation Strategies

Software implementations on powerful parallel computer systems of the elastic net and Potts algorithms facilitated the extensive numerical studies reported in this thesis. This appendix summarizes the key aspects of how the neural algorithms were mapped onto parallel architectures. A basic version of the elastic net implementation was developed as part of an Honours year project by the author.

DAP Implementation of the Elastic Net Algorithm

As the beads which model the elastic net are all updated in parallel, the algorithm is naturally suitable for implementation on Single-Instruction- Multiple-Data (SIMD) computers, such as the AMT-DAP or other massively parallel array processors. The DAP can be considered as a 2-D lattice of processors connected by nearest neighbour communications links, with each processor having a small amount of local memory. From equations 2.1 and 2.2, the computation of the city-bead force terms is clearly the dominant operation in each update. Therefore, an efficient mapping should be optimized with respect to the computation

of the city-bead distance terms and the weight terms derived from these.

There are basically two ways to map the algorithm onto this architecture. Either by associating each bead with a particular processor, storing the set of city coordinates in each local memory, or by associating each city with a particular processor, the local memory of each processor holding the net coordinates. With regard to the important issue of minimizing costly inter-processor communications, neither strategy is clearly superior, since the update operation involves global summation steps over both the cities (equation 2.1) and the beads (equation 2.2). However, the former strategy emerges as the more efficient one with respect to another criterion, namely, minimizing memory requirements.

In the 'beads mapping', the displacement of the net due to a single city is computed in parallel for every bead (processor). Each processor maintains a pair of variables which act as accumulators for the coordinate components of the $\sum_i w_{ij}(\boldsymbol{x}_i - \boldsymbol{y}_j)$ sum for the bead (j) that processor is responsible for. In the alternative 'cities mapping', all of the un-normalized city force terms acting on a single bead are computed in parallel. However, because of the normalization step (equation 2.2), each processor requires an extra $2M$ words of memory temporarily to store the components of the M city-bead force terms it computes, prior to normalization.

In its original form, the beads mapping fully utilises the available processors only when their number matches the number of beads. For the TSPs studied here, having at most 200 cities, it would have been inefficient, since the DAPs used for this work had either 1024 or 4096 processors. The mapping was generalized to overcome this problem, by associating each bead with a small number (roughly the ratio of processors to beads) of processors, each processor now being responsible for the interactions of that bead with a subset of the cities. Interprocessor communication steps gather the partial sums for each bead, and scatter the updated position back to the appropriate processors.

Transputer Implementation of the Mean Field Potts Algorithm

The mean field Potts TSP network was implemented on a Meiko Computing Surface, a configurable array of T800 transputers. Each transputer is a fairly powerful processor in its own right, capable of executing its own programme independent of the other transputers. A small number of fast communications links (for the T800, four) are designed into each transputer. This enables a group of transputers to cooperate, via message-passing, in running a single programme. The user is free to configure the links such that the resulting processor topology suits the pattern of inter-processor communications required by the particular application. In general terms, efficient implementations distribute the workload evenly across the processors whilst keeping inter-processor communications to a minimum.

To programme the dynamics specified by equations 4.9 and 4.13, each transputer is assigned to update a distinct subset of the V_{ia} variables. The key implementation decision is whether each transputer's dataset should correspond to all of the variables associated with a subset of the cities, or all those associated with a subset of the partitions. Let us consider the first approach. The dominant step in the update of V_{ia} is the evaluation of $\sum_j d_{ij}(V_{j(a+1)} + V_{j(a-1)})$. With the cities (indexed by j) distributed over the processors, a considerable amount of inter-processor communications would be required to gather the relevant data into the processor, suggesting that this would be an inefficient mapping. Furthermore, this mapping would be inappropriate for dynamics in which the cities are updated serially, since, unless one resorted to some complex programming, only one processor would be doing computation at any time.

The second strategy, distributing the mean field variables associated with a single city across all the processors, has neither of the above drawbacks. The processors are configured in a ring and adjacent processors assigned to adjacent groups of partitions. Formally, processor $k\,(k = 0 \ldots P-1)$ updates

$\{V_{ia} : 1 \leq i \leq N,\ kN/P + 1 \leq a \leq (k+1)N/P\}$, where — for simplicity — it is assumed that N is a multiple of P. Under this mapping each processor can compute almost all of its U_{ia} terms by operating on *local* data — only for its extremal a values does a processor require data from other processors, and even then only from its immediate neighbours. Thus the dominant computational step is carried out efficiently. The subsequent normalization operation to derive the f_{ia} terms (see equation 4.9) requires global communication on the ring. However, the time spent on these communications is small relative to the time spent on calculations, thus they do not seriously impair the programme's efficiency. This statement can be formalised by considering the 'fractional communication overhead' measure, f_C, introduced by [Fox *et al.* 1988]. f_C is the ratio of the time spent doing communications to that spent on useful calculations by a parallel programme.

Analysis of the algorithm's key steps allowed f_C expressions to be derived for both the serial and parallel updating modes. The results quoted below are in terms of the (problem size)/(machine size) ratio (i.e. N/P), denoted by m.

Serial Dynamics $\quad f_C \approx bm^{-2}\left(4 + \frac{1}{P}\right)$

Parallel Dynamics $\quad f_C \approx bm^{-2}\left(\frac{4}{N} + \frac{c}{P}\right)$

b and c are $O(1)$ terms dependent on details of the implementation and certain machine constants (specifically, the time for a floating point operation and times characterising the performance of the communications links). These results imply that for both types of dynamics this implementation is *scalable*, in the sense that increasing the problem size does not increase the fraction of time spent doing concurrent communications, provided the growth in N is matched by growth in the machine size such that m is kept constant. Indeed, for the parallel dynamics code f_C actually tends to 0 as $N \to \infty$, for constant m. Finally, a less demanding efficiency scaling criterion, namely that f_C should vanish as $N \to \infty$ for constant P, is easily satisfied by either type of dynamics.

APPENDIX B

Theory For Random Metric TSPs

The 'distance' matrix for this class of problems contains symmetric off-diagonal elements, each of which is a random variable drawn independently from a flat $[0, 1]$ distribution. Clearly the 'cities' are no longer embedded in some geometrical space, nor in fact does the triangle inequality generally hold. These properties make this random metric class of TSP a severe test for all heuristic methods: even the renowned Lin-Kernighan algorithm shows performance degradation for increasing N [Johnson 1990].

Although the T_c analyses presented in both Chapter 4 and in PS concentrated on a particular class of Euclidean TSPs, much of that work is sufficiently general to apply to random metric TSPs as well. The key difference betweeen the two cases concerns the eigenvalue spectrum of $A^{(k)}$. For the random metric problems, the ζ parameter characterising the spread of the λ_1^k eigenvalues is empirically found to be independent of N, whereas previously, recall, it scaled as $\sqrt{N}$. A numerical study of many random TSP matrices indicates that in the large N limit, ζ lies in the range 1.95–2.02. A value of 1.9 was used in the section 5.3.2 simulations, as that is a typical value for TSPs of the size studied therein. Analytic evaluation of the mean (d) and standard deviation (σ_d) of the

off-diagonal elements in the $\{d_{ij}\}$ matrix is straightforward in the $N \to \infty$ limit, since the elements are then uncorrelated with each other. The values are 0.5 and $1/(2\sqrt{3}) \approx .289$ respectively. With these values for ζ, d and σ_d, the expressions given in (4.26) for $\lambda_{\max}$ and $\lambda_{\min}$ still hold, with the usual caveat that the $\lambda_{\max}$ expression is valid provided that α exceeds some threshold, α', where now

$$\begin{aligned} \alpha' &= 2(d - \zeta\sigma_d/\sqrt{N}) \\ &\to 2d \qquad \text{for large } N. \end{aligned}$$

The synchronous updating T_c equations (4.27–4.30) remain valid, though of course β_o now has a different scaling. The result of this is that γ_c, the maximum feasible step-size when using parallel updating, is now (correct to leading terms) given by

$$\begin{aligned} \gamma_c &= \frac{4\zeta\sigma_d}{\sqrt{N}(2d+\alpha) + 2\zeta\sigma_d} \\ &\to 0 \qquad \text{as} \quad N \to \infty. \end{aligned}$$

This result has been verified in numerical studies similar to those performed for Figure 4.4. It indicates that the computational burden of using small γ rules out using parallel updating dynamics for large random metric TSPs. T_c for serial updating (and the parallel case, provided $\gamma \leq \gamma_c$) with practical β values, is

$$T_c = \frac{\beta + 2d + 2\zeta\sqrt{N}\sigma_d}{N}$$

So T_c is expected to scale as $N^{-1/2}$ for random distance TSPs.

Regarding suitable α, the arguments presented in sections 5.1.1 and 5.1.2 are essentially independent of the distance matrix, and the section 5.1.3 argument now suggests that α ought to be taken larger than $2d$, for large N. For β, section 5.2.1 suggested a value of $(T \ln \rho\mathcal{G} + 2d')$, where $\mathcal{G}$ denotes $\sum_{j \neq i} \exp((-d_{ij-1} - d_{ij+1})/T)$. Because the d_{ij} are drawn independently, it is possible to evaluate $\mathcal{G}$ exactly in the $N \to \infty$ limit, by

$$\begin{aligned} \lim_{N\to\infty} \mathcal{G} &= N \int_0^1 \int_0^1 e^{-(w+z)/T} \mathrm{d}w\, \mathrm{d}z \\ &= NT^2 \left(1 - e^{-1/T}\right)^2 \end{aligned}$$

This result is of limited practical use for finite N (where small correlations inevitably creep into the distance matrix), but we shall ignore this, as we are only interested in the large N scaling behaviour. By plugging this result into the β expression above, taking $T = T_c$ (reasonable in light of the argument in section 5.3.2 that suggested T_f would be close to T_c for random TSPs), simplifying, and keeping only the dominant term, we find

$$\beta \approx \frac{c \ln \rho c^2}{\sqrt{N}} \qquad \text{where } c \equiv 2\zeta\sigma_d.$$

Thus β may be chosen increasingly small as N grows large, so from an energetics perspective the algorithm is definitely scalable for random non-Euclidean TSPs. The β expression above has not been thoroughly tested experimentally, but a brief investigation did suggest that the minimal value of β required to give convergence to tours decreases as N grows.

Bibliography

[Aiyer *et al.* 1990] Aiyer, S.V.B., Niranjan, M., and Fallside, F. A Theoretical Investigation into the Performance of the Hopfield Model. *IEEE Transactions on Neural Networks* **1** (2), 204-215 (1990).

[Aiyer 1991] Aiyer, S.V.B. *Solving Combinatorial Optimization Problems using Neural Networks with Applications in Speech Recognition*. PhD Thesis, University of Cambridge, Cambridge (1991).

[Angéniol *et al.* 1988] Angéniol, B., de la Croix Vaubois, G., and le Texier, J-Y. Self-Organizing Feature Maps and the Travelling Salesman Problem. *Neural Networks* **1**, 289-293 (1988).

[Aue 1990] Aue, A. *A new Class of Algorithms for the Steiner Problem*. MSc Thesis, Department of Computer Science, University of Edinburgh (1990).

[Baskaran *et al.* 1986] Baskaran, G., Fu, Y., and Anderson, P.W. On the Statistical Mechanics of the Traveling Salesman Problem. *Journal of Statistical Physics* **45**, 1-25 (1986).

[Bilbro *et al.* 1992] Bilbro, G.L., Snyder, W.E., Garnier, S.J., and Gault, J.W. Mean Field Annealing: A Formalism for Constructing GNC-Like Algorithms. *IEEE Transactions on Neural Networks* **3** (1), 131-138 (1992).

[Beardwood *et al.* 1959] Beardwood, J., Halton, J.H., and Hammersley, J.M. The shortest path through many points. *Proceedings of the Cambridge Philosophical Society* **55**, 299-327 (1959).

[Bonomi & Lutton 1984] Bonomi, E. and Lutton, J-L. The *N*-City Travelling Salesman Problem: Statistical Mechanics and the Metropolis Algorithm. *SIAM Review* **26**, 551-568 (1984).

[Brady 1985] Brady, R.M. Optimization strategies gleaned from biological evolution. *Nature* **317**, 804-806 (1985).

[Brandt *et al.* 1988] Brandt, R.D., Wang, Y., Laub, A.J., and Mitra, S.K. Alternative Networks for Solving the Traveling Salesman Problem and the List-Matching Problem. *Proc. 1988 IEEE International Conference on Neural Networks, Vol. II*, 333-340.

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

[Durbin *et al.* 1989] Durbin, R., Szeliski, R., and Yuille, A. An analysis of the elastic net approach to the traveling salesman problem. *Neural Computation* **1**, 348-58 (1989).

[Fox *et al.* 1988] Fox, G.C., Johnson, M.A., Lyzenga, G.A., Otto, S.W., Salmon, J.K., and Walker, D.W. *Solving Problems On Concurrent Processors: Volume I.* Prentice-Hall, London (1988).

[Frean 1990] Frean, M.R. *Small Nets and Short Paths: Optimising Neural Computation.* PhD Thesis, University of Edinburgh, Edinburgh (1990).

[Fritzke & Wilke 1991] Fritzke, B. and Wilke, P. FLEXMAP – A neural network for the travelling salesman problem with linear time and space complexity. *Proc. 1991 IEEE International Joint Conference on Neural Networks, Vol. III*, 929-934.

[Fu & Anderson 1986] Fu, Y. and Anderson, P.W. Application of statistical mechanics to NP-complete problems in combinatorial optimisation. *Journal of Physics A: Mathematical and General* **19**, 1605-1620 (1986).

[Garey & Johnson 1979] Garey, M.R. and Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco (1979).

[Gislén *et al.* 1989] Gislén, L., Peterson, C. and Söderberg, B. "Teachers and Classes" with Neural Networks. *International Journal of Neural Systems* **1**, 167-176 (1989).

[Goldberg 1989] Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading Massachusetts (1989).

[Goodhill 1992] Goodhill, G.J. *Correlations, Competition, and Optimality: Modelling the Development of Topography and Ocular Dominance.* PhD Thesis, University of Sussex, Brighton (1992).

[Goodhill & Willshaw 1990] Goodhill, G.J. and Willshaw, D.J. Application of the elastic net algorithm to the formation of ocular dominance stripes. *Network* **1**, 41-59 (1990).

[Hegde *et al.* 1988] Hegde, S.U., Sweet, J.L., and Levy, W.B. Determination of parameters in a Hopfield/Tank computational network. *Proc. 1988 IEEE International Conference on Neural Networks, Vol. II*, 291-298.

[Held & Karp 1970] Held, M. and Karp, R.M. The Traveling-Salesman Problem and Minimum Spanning Trees. *Operations Research* **18**, 1138-1162 (1970).

[Hérault & Niez 1989] Hérault, L. and Niez, J-J. Neural Networks and Graph K-Partitioning. *Complex Systems* **3**, 531-575 (1989).

[Hopfield 1982] Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Science USA* **79**, 2554-2558 (1982).

[Hopfield 1984] Hopfield, J.J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Science USA* **81**, 3088-3092 (1984).

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

[Jacobs 1988] Jacobs, R.A. Increased Rates of Convergence Through Learning Rate Adaptation *Neural Networks* **1**, 295-307 (1988).

[Jaynes 1957] Jaynes, E.T. Information Theory and Statistical Mechanics. *Physical Review* **106**, 620-630 (1957).

[Johnson *et al.* 1989] Johnson, D.S., Aragon, C.R., McGeoch, L.A., and Schevon, C. Optimization by Simulated Annealing: An Experimental Evaluation, Part I (Graph Partitioning). *Operations Research* **37**, 865-892 (1989).

[Johnson 1990] Johnson, D.S. Local Optimization and the Traveling Salesman Problem. In *Proc. 17th Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science **443**, 446-461, Springer Verlag, Berlin (1990).

[Johnson *et al.* 1992] Johnson, D.S., Aragon, C.R., McGeoch, L.A., and Schevon, C. Optimization by Simulated Annealing: An Experimental Evaluation, Part III (The Traveling Salesman Problem). In preparation.

[Kastella 1992] Kastella, K. Control Parameter Scaling in a Hopfield-Tank List-Matching Network. *Physical Review A* **46** (2), 1116-1123 (1992).

[Kirkpatrick *et al.* 1983] Kirkpatrick, S., Gelatt Jr., C.D., and Vecchi, M.P. Optimization by Simulated Annealing. *Science* **220**, 671-680 (1983).

[Kirkpatrick & Toulouse 1985] Kirkpatrick, S. and Toulouse, G. Configuration space analysis of travelling salesman problems. *Journal de Physique* **46**, 1277-1292 (1985).

[Kohonen 1988] Kohonen, T. *Self-Organization and Associative Memory*. Springer Verlag, Berlin (1988).

[Krauth & Mézard 1989] Krauth, W. and Mézard, M. The Cavity Method and the Travelling-Salesman Problem. *Europhysics Letters* **8** (3), 213-218 (1989).

[Lawler *et al.* 1985] Lawler, E.L., Lenstra J.K., Rinnooy Kan, A.H.G., and Shmoys, D.B. (Eds.) *The Traveling Salesman Problem*. John Wiley & Sons, Chichester (1985).

[Lin 1965] Lin, S. Computer solutions of the traveling salesman problem. *Bell System Technology Journal* **44**, 2245-2269 (1965).

[Lin & Kernighan 1973] Lin, S. and Kernighan, B.W. An Effective Heuristic Algorithm for the Traveling Salesman Problem. *Operations Research* **21**, 498-516 (1973).

[Miller & Pekny 1991] Miller, D.L. and Pekny, J.P. Exact Solution of Large Asymmetric Traveling Salesman Problems. *Science* **251**, 754-761 (1991).

[Mouritsen 1984] Mouritsen, O.G. *Computer Studies of Phase Transitions and Critical Phenomena*. Springer-Verlag, Berlin (1984).

[Mühlenbein *et al.* 1988] Mühlenbein, H., Gorges-Schleuter, M., and Krämer, O. Evolution algorithms in combinatorial optimization. *Parallel Computing* **7**, 65-85 (1988).

[Padberg & Rinaldi 1988] Padberg, M. and Rinaldi, G. A branch and cut algorithm for the resolution of large scale symmetric traveling salesman problems. Report R. 247, Instituto di Analisi dei Sistimi ed Informatica del CNR, Rome (1988).

[Parisi 1988] Parisi, G. *Statistical Field Theory.* Addison-Wesley, Redwood City, California (1988).

[Peterson 1990] Peterson, C. Parallel Distributed Approaches to Combinatorial Optimization: Benchmark Studies on TSP. *Neural Computation* **2**, 261-269 (1990).

[Peterson & Anderson 1987] Peterson, C. and Anderson, J.R. A Mean Field Learning Algorithm for Neural Networks. *Complex Systems* **1**, 995-1019 (1987).

[Peterson & Anderson 1988] Peterson, C. and Anderson, J.R. Neural networks and NP-complete optimization problems; a performance study on the graph bisection problem. *Complex Systems* **2**, 59-89 (1988).

[Peterson & Söderberg 1989] Peterson, C. and Söderberg, B. A New Method for Mapping Optimization Problems onto Neural Networks. *International Journal of Neural Systems* **1**, 3-22 (1989).

[Ruján 1988] Ruján, P. Searching for optimal configurations by simulated tunneling. *Zeitschrift für Physik B* **73**, 391-416 (1988).

[Rumelhart *et al.* 1986] Rumelhart, D.E., Hinton G.E., and Williams, R.J. Learning Internal Representations by Error Propagation. In D.E. Rumelhart and J.L. McLelland, (Eds.), *Parallel Distributed Processing* (Chap. 8). MIT Press, Cambridge MA (1986).

[Silva & Almeida 1990] Silva, F.M. and Almeida, L.B. Acceleration Techniques for the Backpropagation Algorithm. *Lecture Notes in Computer Science* **412**, 110-119 (1990).

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

[Steele 1990] Steele, J.M. Probabilistic and worst case analyses of classical problems of combinatorial optimization in Euclidean space. *Mathematics of Operations Research* **15** (4), 749-770 (1990).

[Tollenaere 1990] Tollenaere, T. SuperSAB: Fast Adaptive Back Propagation with Good Scaling Properties. *Neural Networks* **3**, 561-573 (1990).

[Tollenaere 1991] Tollenaere, T. A Few Notes on SuperSAB. Unpublished (1991).

[van den Bout & Miller 1988] van den Bout, D.E. and Miller III, T.K. A Traveling Salesman Objective Function That Works. *Proc. 1988 IEEE International Conference on Neural Networks, Vol. II*, 299-303.

[van den Bout & Miller 1990] van den Bout, D.E. and Miller III, T.K. Graph Partitioning Using Annealed Neural Networks. *IEEE Transactions on Neural Networks* **1** (2), 192-203 (1990).

[van Laarhoven & Aarts 1987] van Laarhoven, P.J.M. and Aarts, E.H.L. *Simulated Annealing: Theory and Applications*. D. Reidel, Dordrecht (1987).

[Willshaw & von der Malsburg 1979] Willshaw, D.J. and von der Malsburg, C. A marker induction mechanism for the establishment of ordered neural mappings: its application to the retinotectal problem. *Philosophical Transactions of the Royal Society B* **287**, 203-243 (1979).

[Wilson & Pawley 1988] Wilson, G.V. and Pawley, G.S. On the stability of the travelling salesman problem algorithm of Hopfield and Tank. *Biological Cybernetics* **58**, 63-70 (1988).

[Wu 1982] Wu, F.Y. The Potts model. *Reviews of Modern Physics* **54**, 235-268 (1982).

[Xu & Tsai 1991] Xu, X. and Tsai, W.T. Effective Neural Algorithms for the Traveling Salesman Problem. *Neural Networks* **4**, 193-205 (1991).

[Yuille 1990] Yuille, A.L. Generalized Deformable Models, Statistical Physics, and Matching Problems. *Neural Computation* **2**, 1-24 (1990).