ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Solving Optimization Problems in Communications Networks Using Computational Intelligence

Shi Haixiang

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2005

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research done by me and has not been submitted for a higher degree to any other University or Institute.

....

Date Shi Haixiang

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

To my families,

who always give me love and support.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Acknowledgments

First of all, I would like to express my deepest appreciation and sincere gratitude to my supervisor, Associate Professor Lipo Wang, for his guidance, invaluable advices and ideas in my study over the past three years in Nanyang Technological University. It is he who gave me strong support and sincere encouragement throughout the tough times of the research. His keen thoughts impress me so much and I think it will benefit me in my future research and work. He helps me a lot in problem solving and research orientation with his broad and profound knowledge in physics, mathematics, artificial intelligence, chaos theory, and combinatorial optimization. Within these years, I not only learned specialized knowledge from him, but also learned the spirit of exploring science. This research work will not be possible without his guidance, and encouragement. The completion of this thesis would be impossible without his close supervision and devotion to education.

I take this opportunity to thank Nanyang Technological University for rewarding me the research scholarship and the staffs in Information Systems Research Lab and Software Engineering Lab, for their technical help and encouragement.

Special thanks go to my families, who always support me and encourage me with their great patience whenever and wherever.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Summary

The fast growth of technology in communication networks greatly changes our life, and communications is one of the most important growth sectors in all leading economies. Combinatorial optimization problems (COPs) arising naturally in communication networks have acquired essential attentions in research and engineering nowadays. Unfortunately, the optimal solution of a large class of these problems requires an amount of computation which increases exponentially with the problem size. In order to tackle these problems, heuristics are often adopted to find optimal or near-optimal solution in a reasonable computational time due to the NP-hard nature of these COPs.

Neural networks are novel and potentially powerful heuristics for this kind of problems because they are intrinsically parallel systems with significant potential for fast hardware implementation. Unfortunately, the past models of neural networks for solving the combinatorial optimization problems are essentially Hopfield-based networks. The original Hopfield neural network (HNN) needs to construct the energy function related to the optimization problem, and implement a gradient descent dynamics to search the optimal solution. The monotonously descent dynamics makes the network easy to be trapped in local minima and results in poor and infeasible solutions. Recently, there have been extensive research interests and efforts in theory and applications of chaotic neural networks (CNN) which have a richer spectrum of dynamic behaviors, such as stable fixed points, periodic oscillations, and chaos. A revised version of the CNN called transiently

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

chaotic neural network (TCNN) implements chaotic behaviors as in the CNN and gradually decreasing the self-coupling which stabilizes the network. However, the TCNN has completely deterministic dynamics and is not guaranteed to settle down at a global optimum no matter how slowly the annealing parameter (the neuronal self-coupling) is reduced.

In this thesis, firstly a new neural network model called the noisy chaotic neural network (NCNN) is presented to solve the topological optimization problem (TOP) in backbone network design. The NCNN employs the chaotic simulated annealing and decaying noise, which make the networks have both stochastic wandering and efficient chaotic searching ability. Experimental results show that the NCNN outperforms HNN in both solution quality and convergence rate. Furthermore, two extensions of the NCNN model, i.e., the gradual noisy chaotic neural network (G-NCNN) and the noisy chaotic neural network with variable threshold (NCNN-VT), are proposed which aim to separate the problem's constraints from the its objective functions. The G-NCNN is implemented by combining the gradual expansion scheme with the NCNN. In this model, the number of neurons in the network are not fixed. On the contrary, it is gradually activated in groups during the searching process. The grouping and sorting of activated neurons are based on the objectives of the optimization problem. The NCNN-VT employs the adaptive mapping scheme with the NCNN model. It maps the objective function of optimization problem into the adaptive selection of neuron firing probability, i.e., the neuron in the NCNN-VT are assigned with a probability value in the neuron computation, the larger the value of neuron the more likely the neuron be selected in the final solution. With these two schemes combined with the NCNN, the objectives can be separated from the constraints terms. Experimental results show that the proposed neural network models need lower computation cost compared with the previous methods with the growing of the problem size. So our proposed neural networks are more scalable than previous techniques. Moreover, the separation makes the formulation of energy function of neural network simpler and

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

the tuning of weighting coefficients between constraints easier. In this thesis, the convergence and feasibility of the proposed models are provided, together with the analysis of the computational complexity of our models in one iteration. We also discussed the selection of parameters by investigating the single neuron dynamics of the neural network model.

The wide applicability of these neural network models is demonstrated in this thesis through the applications of the three practical COPs in communication networks: topological optimization problem (TOP) in network design, broadcast scheduling problem (BSP) in packet radio networks, and frequency assignment problem (FAP) in satellite communications systems. Each of these problems are solved by different proposed method and compared with the existing algorithms in each area. Besides the comparisons with the previous methods in benchmark examples, extensive randomly generated large problems are solved. The comparative results of solution quality show that the NCNN and the proposed neural network models are attractive methods for solving the combinatorial optimization.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Contents

abstract **i**

Table of Contents **iv**

List of Figures **ix**

List of Tables **xiv**

1 Introduction **1**

1.1 Motivation 1

1.2 Objective 8

1.3 Major contributions of the thesis 9

1.4 Organization of the Thesis 10

2 The Noisy Chaotic Neural Network for Combinatorial Optimization **13**

2.1 Introduction 13

2.2 The Hopfield Model 14

2.2.1 The Mathematical Formulation 14

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

2.2.2 Stability Analysis of the HNN 16

2.3 Transiently Chaotic Neural Network 17

2.3.1 Stochastic Simulated Annealing 17

2.3.2 Chaotic Simulated Annealing 18

2.3.3 Mathematical Formulation 20

2.4 Noisy Chaotic Neural Network 23

2.4.1 Stochastic Chaotic Simulated Annealing 23

2.4.2 Mathematical Formulation 24

2.4.3 Single Neuron Dynamics of the NCNN 25

2.4.4 Stability Analysis for the NCNN 31

2.4.5 Computational Complexity of the NCNN 34

2.5 Concluding Remarks 34

3 Gradual Noisy Chaotic Neural Network and Noisy Chaotic Neural Network with Variable Threshold 37

3.1 Introduction 37

3.2 Gradual Noisy Chaotic Neural Network 38

3.2.1 Gradual Expansion Scheme 38

3.2.2 Discussions on the G-NCNN 40

3.3 Noisy Chaotic Neural Network with Variable Threshold 42

3.3.1 Adaptive Mapping Scheme 42

3.3.2 Model Definition 43

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

3.3.3 Mapping Functions 45

3.4 Parameter Discussion 48

3.5 Concluding Remarks 49

4 Solving the Topological Optimization Problem Using the Noisy Chaotic Neural Network 51

4.1 Introduction 51

4.2 Literature Review on the TOP 53

4.2.1 Calculation of ATR 53

4.2.2 Previous Algorithms on solving this problem 54

4.3 Problem Formulation 55

4.4 Applying NCNN to Topological Optimization Problem 57

4.4.1 Neural Network Formulation 57

4.4.2 Energy Function 59

4.5 Simulation Results 60

4.5.1 Selection of Network Coefficients 60

4.5.2 Benchmark Instances and Results 61

4.6 Conclusions 69

5 Solving the Broadcast Scheduling Problem in Wireless Multihop Networks Using Hybrid Neural Networks 70

5.1 Introduction 70

5.2 Previous Methods on Solving the BSP 71

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

5.3 The Broadcast Scheduling Problem 73

5.4 Hybrid Neural Network Method for the BSP 76

5.4.1 Phase I: Minimize TDMA frame length using SVC 76

5.4.2 Phase II: Maximize the channel utilization 79

5.5 Simulation Results 80

5.5.1 Parameter Selection 80

5.5.2 Evaluation Indices 81

5.5.3 Benchmark Problems 82

5.6 Conclusions 89

6 Solving the Broadcast Scheduling Problem in Wireless Multihop Networks Using the Gradual Noisy Chaotic Neural Network 92

6.1 Introduction 92

6.2 A Gradual Noisy Chaotic Neural Network for the BSP 93

6.2.1 Phase I: Minimizing the TDMA Frame Length Using a G-NCNN 93

6.2.2 Phase II: Maximizing Node Transmissions 95

6.3 Simulation Results 97

6.3.1 Parameter Selection 97

6.3.2 Benchmark Problems 97

6.3.3 Result Discussions 99

6.4 Conclusions 116

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

7 Solving the Frequency Assignment Problem in Satellite Communications Systems Using the Noisy Chaotic Neural Network with Variable Threshold 120

7.1 Introduction 120

7.2 Problem Description 124

7.3 Existing Techniques for the FAP 126

7.4 Application of the NCNN-VT to the FAP 128

7.4.1 Neural network formulation 128

7.4.2 The Mapping Function 130

7.4.3 Energy function 132

7.4.4 Computational Complexity of the NCNN-VT 137

7.5 Computational Results and Comparisons 137

7.5.1 Parameter Discussion 137

7.5.2 Simulation Problems 138

7.5.3 Result Discussions 141

7.6 Concluding remarks 148

8 Conclusion and Recommendations 149

8.1 Conclusions 149

8.2 Recommendations for Future Research 152

Bibliography 156

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

List of Figures

2.1 The input-output function of the HNN with different gain α. 15

2.2 The activation function with different values of ε. 21

2.3 The neuro-dynamics in the single neuron model with different parameter k. $\alpha = 0.015$, $\beta_1 = 0.001$, $\beta_2 = 0.0001$, $\varepsilon = 1/250$, $I_0 = 0.65$, $z(0) = 0.08$, and $n(0) = 0.001$. 28

2.4 The neurodynamics in the single neuron model with different parameter $z(0)$. $k = 0.9$, $\alpha = 0.015$, $\beta_1 = 0.001$, $\beta_2 = 0.0001$, $\varepsilon = 1/250$, $I_0 = 0.65$, and $n(0) = 0.001$. 29

2.5 The neurodynamics in the single neuron model with different parameter $n(0)$. $k = 0.9$, $\alpha = 0.015$, $\beta_1 = 0.001$, $\beta_2 = 0.0001$, $\varepsilon = 1/250$, $I_0 = 0.65$, and $z(0) = 0.08$. 30

3.1 An example of the gradual expansion scheme in the G-NCNN. (a) The cost. (b) The multi-stage activation of the G-NCNN by adopting the gradual expansion scheme. The black squares indicate the newly added neurons in the stage and the dark gray squares are consecutively activated neurons in the past stages. 41

3.2 The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.1$. 43

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

3.3 The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.3$. 44

3.4 The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.6$. 44

3.5 The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.9$. 45

3.6 Different mapping functions for the NCNN-VT. (a) a linear mapping function. (b) a nonlinear mapping function, case 1. (c) a nonlinear mapping function, case 2. 47

4.1 A neural network formulation for the optimal solution of benchmark problem 1 (fully connected): (a) the final optimal network with 5 vertices and 7 edges, (b) the $N \times N$ neural network representation for an N-vertex network, and (c) the final neural network representation. 58

4.2 Comparison of the computation time required by the proposed method and the previous methods. The x-axis represents the cases from 1 to 20 (the problem size increases from case 1 to case 20), and the y-axis is the computation time required by each case. 63

4.3 Error bars for the cost of the randomly generated network instances. The x-axis stands for each instance and the y-axis is the network cost. 68

5.1 Situations in which conflicts occur in a packet radio network. (a) Primary conflict. Node i should not have transmission and reception simultaneously. (b) Secondary conflict. Node k should not receive two or more transmissions simultaneously. 74

5.2 The packet radio networks of the three benchmark instances used in [112]. (a) 15-node-29-edge, (b) 30-node-70-edge, and (c) 40-node-66-edge. 83

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

5.3 The broadcast schedule for the 40-node-66-edge instance. N and M stand for the number of nodes and the number of time slots, respectively. The black square stands for the transmission of node i in slot j. 85

5.4 Comparisons of average time delays for the the three benchmark problems BM 2 and BM 3 (according to Pollaczek-Khinchin formula eqn. (5.16)) with different approaches. 86

5.5 Comparison of the computation time required by the proposed method and the previous methods. The x-axis represents the BM 1 to 3, and the y-axis is the computation time required by each case. 88

6.1 A neural network formulation for a broadcast scheduling problem. . 94

6.2 The optimal TDMA frame obtained using the G-NCNN on benchmark instance BM 1. 100

6.3 Same as Fig. 6.2 for BM 2. 101

6.4 Same as Fig. 6.2 for BM 3. 102

6.5 Noisy chaotic dynamics of the neural network energy for benchmark BM 2. 103

6.6 A comparison of the average time delay as a function of the total arrival rate for the 15-node-29-edge benchmark (according to the Pollaczek-Khinchin formula given by eqn. (28)) among different approaches. 104

6.7 Same as Fig. 6.6, for the 30-node-70-edge benchmark. 105

6.8 Same as Fig. 6.6, for the 40-node-66-edge benchmark. 106

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

6.9 Comparisons of channel utilization for the three benchmark problems. 1, 2, and 3 in the horizontal axis stand for benchmark problems BM 1, BM 2, and BM 3 in Table 1, respectively. 107

6.10 Comparison of the computation time required by the proposed method and the previous methods. The x-axis represents the BM 1 to 3, and the y-axis is the computation time required by each case. 111

6.11 Error bars for the average delay time η given by eqn. (26) obtained using the G-NCNN for 20 randomly generated cases of the BSP. The results are average values obtained through 30 different runs. Since the standard deviations for the GNN are not available, only the best (minimum) results for the GNN are plotted here (x-marks). 115

7.1 Inter-system co-channel interference. 121

7.2 Intra-system co-channel interference, where the B_I stands for the beam I. B_j, B_k, and B_L have the same meanings. 122

7.3 Co-channel interference model for the system in Fig. 7.1, where the C_{xy} stands for the carrier y in system x, e.g., C_{23} stands for the carrier 3 in system 2. 124

7.4 Segmentation and interference matrix for the system in Fig. 7.3. (a) Segmentation of the two systems. (b) the interference matrix $E = (e_{ij})$ for the M-segment system, where the S_{xy} stands for the y-th segment in system x. 125

7.5 The neural network formulation for the FAP. (a) The 24 neurons for the 4-carrier-6-segment FAP. (b) the convergence state of the neural network. (c) the full assignment of the neural network formulation. (d) the final assignment of the segments for the FAP through the expansion from the neural network formulation. 129

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

7.6 Cost computation for each neuron from interference matrix E. (a) The computation method, where the lines in (a) show the crossed elements which are needed in computing the maximum cost. (b) The final cost matrix obtained using the computation method. . . . 131

7.7 Probability matrix of neuron firing by different mapping functions (a) cost matrix. (b) probability matrix obtained by linear mapping in eqn. (7.1). (c) probability matrix obtained by nonlinear mapping in eqn. (3.4), and (d) probability matrix obtained by nonlinear mapping in eqn. (3.6). 133

7.8 Violation condition for the second and third constraints of the FAP. 134

7.9 The interference matrix and the best solution for BM 5. 141

7.10 The best assignment achieved by the NCNN-VT for BM 5. 142

7.11 The computation time required by the NCNN-VT for the test instances. The x-axis represents the instances from 1 to 20, and the y-axis is the computation time required by each instance. 147

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

List of Tables

4.1 Comparisons of network cost obtained using the NCNN for benchmark examples from [31] with previous methods, where the results for the NCNN is listed as best cost, average cost, and standard deviation, (Best/Avg/SD) in 50 runs. 62

4.2 Comparisons of average computation time (seconds) for benchmark examples with previous methods in 50 runs. 64

4.3 Results of randomly generated network instances 65

4.4 List of the weighting coefficients for simulated problems 66

5.1 Specifications of benchmark instances. 82

5.2 Comparisons of best and average results obtained by our SVC-NCNN ($W_1 = 1.0$ and $W_2 = 0.6$) with different algorithms in the three benchmark problems in 50 runs, where η is the average time delay, M is the minimal time slots, and T is the computational time needed (in seconds). 87

5.3 Simulation results for benchmark example 4 to 6 using our SVC-NCNN on a desktop personal computer with a 2.4GHZ CPU and various choices of W_2 (W_1 is fixed at 1). Here convergence rate is defined as the ratio of the runs in which feasible solutions can be found to all number of runs. 90

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

6.1 Parameters of the NCNN model for the simulated instances with $W_1 = 1.0, W_2 = 1.0$, and $W_3 = 1.0$. 98

6.2 Specifications of the three benchmark examples given by [112] (BM 1 - 3) and the geometric instances randomly generated as in [36] (cases 1 to 20). The lower bounds for the number of time slots M are displayed as average/maximum/minimum values. 108

6.3 Comparisons of average delay time η and numbers of time slots M and computation time T obtained by the G-NCNN and other algorithms for the three benchmark problems in 50 runs, where Best/Avg stands for the best value and average value in multiple runs. 110

6.4 A paired t-test of average time delay η (second) between the HNN-GA and the G-NCNN. 112

6.5 Results of the TCNN and the NCNN in the 100-node 522-edge benchmark instance using various noise levels in 20 different runs. Here convergence rate is defined as the ratio of the runs in which feasible solutions can be found to all number of runs. 113

6.6 Same as Table 6.5 for the 250-node 1398-edge benchmark instance. Here convergence rate is defined as the ratio of the runs in which feasible solutions can be found to all number of runs. 114

6.7 Same as Table 6.5 for the 750-node 4371-edge benchmark instance. Here convergence rate is defined as the ratio of the runs in which feasible solutions can be found to all number of runs. 116

6.8 Comparisons of average delay time η given by eqn. (26) and time slot M obtained by the G-NCNN and the GNN on the randomly generated geometric graph instances. The results are displayed as average $\pm$ standard deviation / maximum / minimum. 117

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

6.9 The computation time for the benchmark instances run on a personal computer with a P4 2.4 GHz CPU (average±standard deviation) seconds 118

7.1 Parameters of the NCNN-VT model for the benchmark examples, with $W_1 = 1.0$. 139

7.2 Specifications of the benchmark FAP used in the simulations. 140

7.3 Comparisons of the simulation results (largest interference and total interference) obtained by the NCNN-VT, GNN and HopSA in the benchmark examples. 142

7.4 Comparisons of the NCNN and the NCNN-VT on the benchmark examples with 50 different runs, where η is the convergence rate, T is the average number of interation steps, I_{max} is the largest interference and I_t is the total interference. Here the interference is shown in best and average value (Best/Ave). 143

7.5 Simulation results (largest interference and total interference) on the randomly generated instances between the HopSA [93] and the NCNN-VT. The results are obtained in 50 runs and are displayed as mean±SD (standard deviation). 144

7.6 Iteration steps, computation time (second), and convergence rate for the NCNN-VB on a Linux cluster with 16-node Dual Xeon 3.06 GHz (Intel IA32). The results are obtained through 50 runs and are displayed as mean±SD (standard deviation). Here convergence rate is defined as the ratio of the runs in which feasible solutions can be found to all number of runs. 145

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

List of Acronyms and Abbreviations

ABBREVIATIONS	FULL EXPRESSIONS
ATR	All-Terminal Reliability
AMS	Adaptive Mapping Scheme
NCNN-VT	Noisy Chaotic Neural Network with Variable Threshold
BnB	Branch and Bound
BSP	Broadcast Scheduling Problem
CNN	Chaotic Neural Network
COP	Combinatorial Optimization Problem
CSA	Chaotic Simulated Annealing
FAP	Frequency Assignment Problem
GA	Genetic Algorithm
GES	Gradual Expansion Scheme
G-NCNN	Gradual Noisy Chaotic Neural Network
GNN	Gradual Neural Network

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

HNN	Hopfield Neural Network
MFA	mean field annealing
NCNN	Noisy Chaotic Neural Network
SSA	Stochastic Simulated Annealing
SVC	Sequential Vertex Coloring
TCNN	Transiently Chaotic Neural Network
TOP	Topological Optimization Problem
TSP	Traveling Salesman Problem

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 1

Introduction

1.1 Motivation

Combinatorial optimization problems (COPs) cover a wide spectrum of areas in various fields including economics, biology, physics and engineering. Combinatorial optimization involves searching a large space of possible solutions for a specific optimal solution. The objective of COPs is to find such a specific optimal solution while meet certain constraints and the values of the variables in the final solutions are restrained to be integers, i.e., limited to binary values of 0 and 1. Optimality of a specific solution for COPs is generally measured by an analytical function of the problem, or called goal function or objective function of the problem. The search of optimal solution is to maximize or minimize the objective function, where one solution maybe better than the other. Usually the combinations of the solution are large despite the constraints which restrict the feasible region of solutions. In practical situation, the COPs are constraint satisfaction problems with one or several constraints. When the constraints and objective functions are linear, the global solution is easy to find. Actually in usual case, the objective functions and constraints are nonlinear functions which has many local minima. And to find a global optimal is not easy among vast number of combinations. Such problems

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

are said to be NP-hard since they are unlikely to be solvable by an amount of computational effort which is bounded above by a polynomial function of the size of problem [40].

In order to find an optimal solution among many possibilities, an effective algorithm should be applied. There are several classes of methods to attack a global optimization problem. Usually, exact algorithms or heuristic (approximation) algorithms are two general ways. As an exact method, Branch and bound (BnB) is by far the most widely used tool for solving large scale NP-hard combinatorial optimization problems. It has been widely used in many applications in recent research and engineering fields and it is an algorithm paradigm, which has to be filled out for each specific problem type, and numerous choices for each of the components exist [67, 54, 107, 60, 33, 77, 26]. Even then, principles for the design of efficient BnB algorithm have emerged over the years. Given a specific optimization problem, BnB algorithm searches for the best solution in the complete space of solutions. However, explicit enumeration is normally impossible due to the exponentially increasing number of potential solutions. A search tree is formed which applies node selecting scheme and branching rule to traverse the solution tree. The use of bounds for the function to be optimized combined with the value of current best solution enables the algorithm to search only parts of the solution space implicitly. If the bounding function can drastically reduce the solution space from one tree level to next, the total number of nodes explored will be reduced. Thus we can eventually search efficiently in relatively small iterations and guaranteed to find a global optimal solution. In this thesis, we will compare our results with BnB and show that the exact method is not suitable for the large case combinatorial optimization problems although it is successful to small-size ones.

The computational burden for NP-hard COPs can be illustrated by investigating the most widely studied problem in combinatorial optimization, i.e., the famous traveling salesman problem (TSP) [72]. In this problem, a salesman is

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

about to visit N cities and passes each city once and exactly once while ending his tour in the same city where he started. The inter-city distances for all pairs of cities are known as a priori, i.e., the pre-defined distance matrix specified in the form of an $N \times N$ matrix where N is the number of city. The objective of the TSP is to find the shortest tour among all valid tours without those which invalid paths including may pass through each city more than once, or do not complete the tour in a full circle. The solution space of this problem can be determined to be $N!/2N$ [72]. Let us examine the total solution space through several examples. For a seven-city TSP ($N = 7$), the solution space consists of 360 valid paths. Therefore, it is feasible to calculate the tour length associated with each of the 360 paths and select the shortest one. This solution is called the global optimum solution for the TSP since we have explicitly determined that among all possible valid tours. However, with a slightly larger problem consisting of 20 cities ($N = 20$), the solution space explodes into greater than 6×10^{16} possible valid tours. Even if one could calculate the length of each tour in 100 ns (using a very fast processor) it would require more than 190 years to measure all the possible tour lengths. Obviously such an "exhaustive search" of the solution space is impractical and infeasible, especially since in most real-world applications, the number of cities is of the order of several thousand. The TSP belongs to the NP-complete class of problems, meaning that no polynomial time algorithm has yet been found that can find the global optimum solution. Therefore, most approaches to solving the TSP are heuristic techniques which tend to only sample the solution space in search of an optimal or sub-optimal solution. Since the entire solution space is not searched, there is no guarantee that the chosen good solution is the global optimum. However, in many real-world applications, it is common to prefer solutions that are sufficiently optimal and can be found rapidly over more exhaustive search techniques which can find the global optimum solution but consume valuable computational resources.

From the practical view, heuristic algorithms are more useful than the exhaustive methods and have achieved significant success when applied to a large number

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

of optimization problems. The heuristic algorithms developed so far include greedy methods, local search methods, randomized methods, relaxation techniques, partial enumeration, decomposition and partition approaches, and so on [89] [134]. Among these heuristic algorithms, the local search methods have extensive applications. On the basis of local search, some modern heuristic algorithms are developed, such as simulated annealing (SA) [65], genetic algorithms (GA), Tabu Search [42] [43], and Neural Networks.

The Hopfield neural network (HNN) is a novel and potentially powerful heuristics for this kind of problems because they are intrinsically parallel systems with significant potential for fast hardware implementation [51, 58]. It is a typical model of artificial neural network with symmetric connection weights [50, 51, 52]. It has been shown to be a powerful tool for combinatorial optimization [91, 52, 120, 85, 89, 12, 19, 37, 9, 106, 6]. In [106] Smith summarized the previous work on using the HNN. They have traced the history of neural network research from the perspective of the two main approaches: Hopfield networks and self-organizing networks. A detail survey of applications of neural networks for the various COPs have also been given [106]. Varma and Jayadeva [111] proposed a novel digital neural network for the traveling salesman problem. Recently application of HNN on solving the QoS dynamic connection-admission control in multimedia wireless networks was discussed in [6]. Although the HNN is a powerful tool for solving a wide spectrum of combinatorial optimization problems and guarantees convergence to a stable equilibrium point, it suffers the main shortcoming that it can be easily stuck in the local minima due to its gradient descent dynamics [21]. Recently hill climbing scheme had been used to perform as an excitatory force to enable the network to escape from local minima, e.g. [36]. Wilson and Pawley [123] investigate the HNN and raise the doubts to the reliability and validity of the HNN for solving the TSP. They conclude that the original HNN is unreliable.

A practical way to help the system to get out of local minima is to use noise to occasionally allow the system to jump to states of high energy. Particularly in

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

optimizations where a global search for an optimal solution should be constantly perturbed by noise in order to avoid being trapped into local optimal solutions. In this sense noise in neural networks ensures the network to converge to some global optima of a certain energy function by randomly forcing the network to consider locally non-optimal states. This idea has led to the development of a novel type of optimization strategies generally known as simulated annealing (SA). Simulated annealing is an example of an evolutionary process modeled accurately by purely stochastic means and is believed to be a plausible model of selection processes in nature [90]. That is the process starts at a high temperature where choosing less probable states occurs often and gradually decreases in temperature until changes in states always increase the local probabilities [65]. The generality of the method gives promise that simulated annealing will be a very widely applicable heuristic optimization technique. In recent study, different simulated annealing techniques have been combined with neural networks for optimization problems. A typical example is the Gaussian machines obtained by adding random noises with Gaussian distribution into an analogue neural network, i.e., stochastic simulated annealing (SSA) [8].

It is well know that SSA tends to find high-quality solutions (global optima or near-global optima), if the annealing parameter (temperature) is reduced exponentially but with a sufficiently small exponent. However, for many applications, this may lead to prohibitively long relaxation time in order to find solutions of acceptable quality, and conversely, reasonable long periods of time may still result in poor solutions. In order to speed up the searching process, various research efforts have been made on finding new search technique based on chaotic search. Recently, a large number of chaotic neural networks (CNN) with chaotic dynamics have been investigated [105, 117, 130, 7, 80, 124, 53, 115, 114, 125, 21, 22, 23, 24, 45, 46, 48, 116]. When applied to solving the optimization problem, the chaotic dynamics will boost the searching ability of the neural network without being trapped in local minima.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Firstly, Nozawa demonstrated the search ability of chaotic neural networks [80] [124]. Chen and Aihara [125] [21] proposed a transiently chaotic neural network (TCNN) by introducing a new variable corresponding to the "temperature" in the usual annealing process into chaotic Neural Network (CNN) model to harness the chaotic dynamics. The TCNN was considered as chaotic simulated annealing (CSA) [21] because the dynamics of the TCNN is deterministically chaotic rather than stochastic. Their computer simulations showed that CSA leads to good solutions for the TSP much more easily compared to the Hopfield-Tank approach [50] [51] [52] and SSA [65]. Chen and Aihara discussed the theoretical explanation for the global searching ability of the chaotic neural network in [23]. They claimed that the attracting set of the CNN contains all global and local optima of the optimization problem under certain conditions, and since the chaotic attracting set has a fractal structure and covers only a very small fraction of the entire state space, CSA is more efficient in searching for good solutions for optimization problems compared to other global search algorithms such as SSA, because the stochastic search in SSA covers the entire solution space, rather than a fraction of the solution space covered by the search in CSA.

Besides the CSA in the CNN [125], another type of CNN is obtained by adding the chaotic noise into each neuron of the discrete-time continuous-output HNN [47] [48]. The chaotic noise used can be generated from the logistic map or other chaotic generators. In [47], Hayakawa *et al.* used chaotic noise to the discretized Continuous-time Continuous-output HNN to solve the TSP. The results show the low convergence rate due to the nonconvergent chaotic noise used in their system. He [48] adopted a reduced chaotic noise in their model, as the chaotic noise approaches zero, the network becomes the discrete-time continuous-output HNN, thereby stabilizing and minimizing the energy. The results on solving the TSP shows good convergence rate. In [57], Jayadeva proposed the sequential chaotic annealing neural network (SCA) and applied to multilayer channel routing. In [14], Bhardwaj and Jayadeva applied the SCA to optimal multiuser detection problem

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

in CDMA system.

Although CSA has the feature of searching efficiently, it is unable to guarantee to settle down to a global minima because CSA has completely deterministic dynamics [109]. In [118], Wang and Tian combine the best features of both SSA and CSA and take the advantage of the SSA and CSA, i.e., stochastic chaotic simulated annealing (SCSA), by adding a decreasing noise in the transiently chaotic neural network. A noisy chaotic neural network (NCNN) which has noisy feature and efficient chaotic searching is obtained.

Although the NCNN has been applied to solve the typical combinatorial optimization problems like the TSP and channel assignment problem [113] [116] in GSM network, more practical applications need to be solved using the NCNN to demonstrate its potential. The applicability of the NCNN is not very convincing because it is applied to limited scope of problems and with small-size problems only. In order to demonstrate the applicability of the NCNN model, other problems, especially the new NP-hard COPs arose in communications networks should be solved. Besides, the NCNN should be applied to large-size instances in these problems.

As in recent life, the fast development of communication technology greatly changed our life style. And the increasing demands in communications makes it the most important growth sectors in all leading economies. COPs arise naturally in communications have been widely solved by various methods. The COPs in communications area include scheduling (broadcast scheduling), routing (shortest path routing, unicast/multicast routing), assignment (channel assignment, frequency assignment), and design (topological design) problems. Excessive attentions and research efforts have been put on solving these NP-hard COPs. The applications of the NCNN to these problems are new and original.

Most of the above mentioned NP-hard problems are all belonging to the constraint satisfaction problems, which have one or several goals and many constraints.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

In solving this kind of problem, the scalability of the algorithm is a key factor. Nonscalable algorithms will lead to poor solutions in large difficult problems. We noticed that although the NCNN is a general model, it needs tedious work of parameter selections when applied to problems with growing size. In order to improve the scalability of the NCNN method, new neural network models are worth exploring.

1.2 Objective

The focus of this thesis is on designing efficient heuristic methods for the NP-hard combinatorial optimization problems. Due to the increasing applications and utilizations in communications in science research and industry engineering in recent time, we focus our research problems in communications networks, i.e., we aim to solve NP-hard combinatorial problems in communications networks in recent industrial engineering. Three typical problems are selected in this work, i.e., topological optimization problem (TOP) in network design [100], broadcast scheduling problem (BSP) in packet radio network [98] [99] [102] [101], and frequency assignment problem (FAP) in satellite communications systems [78] [38] [93].

The efficiency of the NCNN in solving the COPs have been demonstrated by the previous work in solving TSP and CAP [116]. In order to show the applicability of the NCNN, we firstly apply the NCNN to other NP-hard COPs mentioned above. With the applications to these problems, we not only demonstrate the performance of our method on already solved benchmark examples, but also on solving randomly generated large-size problems to show the scalability of the NCNN.

Through the applications of the NCNN method to solve the constraint satisfaction NP-hard problems, we found that in order to improve the scalability of the NCNN method, it is necessary to separate the objective terms of the problem from the constraint terms. In this context, we proposed two kinds of original models

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

based on the NCNN model, i.e., gradual noisy chaotic neural network (G-NCNN) and noisy chaotic neural network with variable threshold (NCNN-VT). We apply the G-NCNN to solve the BSP in packet radio networks and use the NCNN-VT to tackle the FAP in satellite communications systems. The simulation results in the applications show that, through the separation of the objectives, the proposed methods can not only simplify the selection of weighting coefficients in the neural network energy function, but also will improve the solution quality in terms of convergence rate and computation time.

While we focused on the design of NCNN-based models, we also attempted to using hybrid methods to tackle the COPs. The idea of hybridization of methods is that it combines the best of the two methods while avoiding the shortcomings. There are several typical hybrid algorithms in the previous research, i.e., Hopfield neural network and genetic algorithm (HopGA) [92] [94] and Hopfield neural network with simulation annealing (HopSA) [93]. In this thesis, we combined the NCNN with the typical heuristic in graph theory, i.e., sequential vertex coloring to solve the BSP.

1.3 Major contributions of the thesis

The original contributions presented in this thesis can be summarized as follows:

1. Applications of the NCNN to three practical NP-hard COPs arising in communications industry and obtaining better solutions than previous methods.

2. Demonstration of the potential of the NCNN model by applying it to the large-size problems.

3. Experimentally proving the superior performance of the NCNN over the TCNN through the comparisons in the application of broadcast scheduling problem.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

4. Proof of the stability and convergence of the NCNN based on a unified framework.

5. Proposal of a new noisy chaotic neural network (G-NCNN) which combines the gradual expansion scheme with the neural network based methods.

6. Proposal of hybrid neural networks which combines the sequential vertex coloring algorithm with the noisy chaotic neural network. The application to the BSP shows its ability through extensive simulation results.

7. Proposal of the adaptive mapping scheme and the original noisy chaotic neural network with variable threshold (NCNN-VT). The NCNN-VT adaptively maps the objective of the problem into the probability of firing for each neurons in the neural network. Simulations of the NCNN-VT on benchmark problems and randomly generated large-size problems are performed.

8. Proposal of a new formulation of the energy function in solving the frequency assignment problem. Discussions on possible problems of the previous formulation are also provided.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows.

Chapter 2 firstly presents the theory of Hopfield neural network and the stability property of the HNN. Then it analyzes the chaotic dynamics of the chaotic neural network, transiently chaotic neural network, and noisy chaotic neural network from a single neuron perspective. The mathematical formulation of these chaotic based neural network are given and the discussion on the difference between them are also presented in the end of this chapter.

In Chapter 3, based on the discussions of the NCNN method, we propose two new models of neural networks which separates the objectives terms from

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

the constraints terms in the problems. The mathematical formulation of these two proposed method are presented. The differences of proposed methods with the NCNN are discussed and the chaotic dynamics of the proposed methods are compared with the NCNN dynamics.

The following four chapters (Chapters 4 to 7) are the detailed applications of our proposed methods in different practical combinatorial problems. In Chapter 4, we consider a network design problem which has its effects in a wide spectrum of areas including telecommunications, computer networks, electricity distribution, gas pipelines, and sewer networks. The problem needs to find an optimal topology of network with minimal costs while meeting the reliability constraint. We apply the NCNN methods to solve the TOP and compare the results with the previous methods.

In Chapter 5, the broadcast scheduling problem (BSP) in packet radio network is solved. The essence of the BSP is to find the optimal TDMA transmission frame with minimal time slot and maximum node transmission in order to make the best of the channel resources. We proposed a hybrid approach which combines the sequential vertex coloring (SVC) algorithm and the NCNN to tackle the BSP in two phases. The SVC is adopted to find minimal number of time slots while meeting the constraints in the first phase. The NCNN is used in the second phase to find additional node transmissions in the second phase. The simulation results show that the performance of our hybrid method outperforms the previous method in this problem.

Chapter 6 is the application of proposed G-NCNN model to the BSP. The G-NCNN solves the BSP in two phases. By the utilization of gradual expansion scheme, the G-NCNN is able to find the minimal number of time slots in the first phase. The second phase is the same as in Chapter 5 by using the NCNN to find additional transmissions while meeting the constraints. Besides the benchmark problems, the G-NCNN model have been applied to 600 different instances with various problem size. We show that the G-NCNN can obtain better solution than

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

the previous methods in all simulated instances.

In order to show the ability of the proposed NCNN-VT model, we apply it to frequency assignment problem in satellite communication systems in Chapter 7, which is also a NP-hard combinatorial problem. The NCNN-VT separates the objective of this problem from the constraints by using the adaptive mapping scheme. The simulation results show that the NCNN-VT is superior to previous methods in both solution quality and computation cost. The proposed NCNN-VT is highly scalable with less work of tuning of network parameters and less computation cost.

We conclude our work in Chapter 8 with the discussions of further research work.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 2

The Noisy Chaotic Neural Network for Combinatorial Optimization

2.1 Introduction

Since Hopfield and Tank's innovative work on solving the TSP using neural networks, there are numerous research efforts on applying the HNN and HNN-based neural network techniques to solving COPs [9, 37, 73, 119, 108, 74, 70]. However, Wilson and Pawley [123] raise the doubts on the validity of the HNN to solving the COPs after they were unable to reproduce the results as in Hopfield and Tank's work. They claimed that the original HNN formulation for the TSP is unreliable even for the small-sized problems. Many explanations for the poor solution quality of TSP had been made in terms of energy function formulation [30] [79] [15] and parameter selection [49] [71] [64] [28].

Besides the dependence on the formulation of energy function and the difficulties in parameter selection, the performance of the original HNN still suffers from

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

its limitation of neuron dynamics. Its gradient descent dynamics leads to the poor solutions which often trapped in local minima, if without using other heuristic schemes like hill climbing. In this chapter, we intend to introduce the chaotic neurodynamics which can help to jump from the local minima and converges to better solutions in solving COPs. After the introduction of the HNN, we will discuss the chaotic simulation annealing and the mathematical formulation of the TCNN. In the end of the chapter, the NCNN with stochastic chaotic simulation annealing and its mathematical models are introduced.

2.2 The Hopfield Model

2.2.1 The Mathematical Formulation

In 1982, Hopfield [50] introduced a neural network model of a Content Addressable Memory (CAM) composed of many highly interconnected two stage McCulloch-Pitts neuron. Since the classical paper of Hopfield and Tank [52], lots of theoretical and experimental research have been published on this subject. Hopfield and Tank [52] applied the HNN to solve the TSP and the impressive results in solving the TSP inspire the boom of research in neural networks. The continuous Hopfield Model composed of N neuronal units, in terms of an electrical circuit, is described by the set of differential equations.

$$C_i \frac{du_i}{dt} = \sum_{j=0}^{N-1} w_{ij} v_j + I_j - \frac{u_i}{r_i} \tag{2.1}$$

where the C_i is the input capacitancy of the cell membrance, and the r_i is the trans-membrane resistance. u_i and v_i denote the input potential and the output potential of the $i-$th neuron, respectively, $i = 0, 1, ..., N-1$, I_i is the external input current to the $i-$th neuron, and $w_{ij} = \frac{1}{R_{ij}}$ is a weight from output of neuron j to input of neuron i. The weights $w_{ij} = w_{ji}$.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

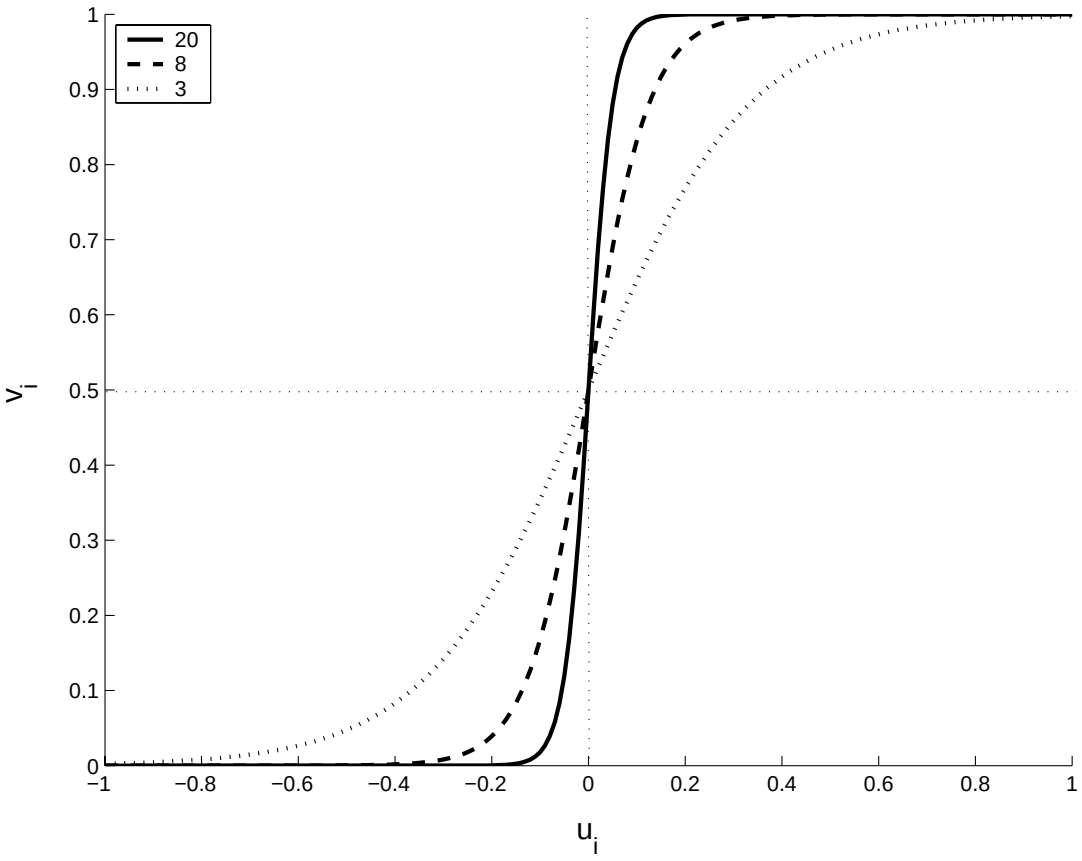


Figure 2.1: The input-output function of the HNN with different gain α.

$$\frac{1}{r_i} = \frac{1}{R_i} + \sum_{j=0}^{N-1} w_{ij} \tag{2.2}$$

The input-output function of the $i-$th neuron, $v_i = g(u_i)$, is a sigmoid function:

$$v_i = g_i(u_i) = \frac{1}{2}(1 + \tanh(\alpha u_i)) \tag{2.3}$$

Fig. 2.1 shows the function in eqn. (2.3) with different gain α.

Solving the given optimization problems with the Hopfield Model is based on minimization of the energy function (Lyapunov function), which has the following generic form:

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} v_i v_j - \sum_{i=1}^{N} I_i v_i + \frac{1}{RC}\sum_{i=1}^{N}\int_0^{v_i} g^{-1}(\eta)d\eta \tag{2.4}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

For sufficiently high gain α, e.g. $\alpha \geq 20$ as in Fig. 2.1, the integral term in the energy eqn.(2.4) vanishes. Thus in the high-gain limit of the input-output function, the continuous Hopfield model reduces to the discrete Hopfield model,

$$E_d = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} v_i v_j - \sum_{i=1}^{N} I_i v_i \tag{2.5}$$

The local minima of discrete Hopfield model lie at the vertices of the unit hypercube and the continuous Hopfield network converges to the 0-1 vertex of the hypercube which minimizes the discrete Hopfield network energy E_d [106].

Therefore, while solving optimization problem with the Hopfield network one is to choose weights w_{ij} and external inputs I_i so as (global) minima of eqn. (2.5) correspond to (optimal) solutions of the given problem at hand.

Commonly the deterministic gradient descent updating rule is used, i.e.,

$$\frac{du_i}{dt} = -\frac{\partial E}{\partial v_i}$$

The convergence to global minimum is not guaranteed and depend on the initial situation.

2.2.2 Stability Analysis of the HNN

In the continuous Hopfield network, the Liapunov function is given as in the form of eqn. (2.4). Given the two conditions [50]:

1. $w_{ij} = w_{ji}$.

2. the input-output function of neurons is monotonously increasing function as in eqn. (2.3).

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Then the time derivative of E is:

$$\begin{aligned}
\frac{dE}{dt} &= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}v_j\frac{dv_i}{dt} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ji}v_j\frac{dv_i}{dt} - \sum_{i=1}^{n} I_i\frac{dv_i}{dt} + \sum_{i=1}^{n}\frac{1}{r_i}u_i\frac{dv_i}{dt} \\
&= -\sum_{i=1}^{n}\frac{dv_i}{dt}\left(\sum_{j=1}^{n} w_{ij}v_j + I_i - \frac{u_i}{r_i}\right) \\
&= -\sum_{i=1}^{n} C_i\frac{dv_i}{dt}\frac{du_i}{dt} \\
&= -\sum_{i=1}^{n} C_i\frac{du_i}{dv_i}\left(\frac{dv_i}{dt}\right)^2 \\
&= -\sum_{i=1}^{n} C_i g_i^{-1'}(v_i)\left(\frac{dv_i}{dt}\right)^2 .
\end{aligned}$$

Since $g_i^{-1}(v_i)$ is a monotonically increasing function, then

$$g_i^{-1'}(v_i) > 0$$

And because C_i is positive, then:

$$\frac{dE}{dt} \leq 0, \qquad \text{and} \qquad \frac{dE}{dt} = 0 \quad \Rightarrow \quad \frac{dv_i}{dt} = 0 \qquad \forall i \; . \tag{2.6}$$

And since the E is bounded, eqn. (2.6) shows that the network energy E will decrease and converge to the minimum.

2.3 Transiently Chaotic Neural Network

2.3.1 Stochastic Simulated Annealing

Since the original Hopfield neural network (HNN) [51] [52] can be easily trapped in local minima, stochastic simulated annealing (SSA) [65] has been combined with the HNN, i.e., the mean field theory algorithm [84] or mean field annealing (MFA) [112]. In 1983, Kirkpatrick *et al* [65] developed simulated annealing, which emulates the solid annealing processing by first heating a solid to its melting point and then slowly cooling the material at a natural rate related to its heat transport

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

speed. Because of its stochastic manner in optimization processing, the annealing mechanics is actually stochastic simulated annealing (SSA) [65]. SSA is known to relax to a global minimum with probability 1 if the annealing takes place sufficiently slowly, i.e., at least inversely proportional to the logarithm of time [41]. In a practical term, this means that SSA is capable of producing good (optimal or near-optimal) solutions for many applications, if the annealing parameter (temperature) is reduced exponentially with a reasonably small exponent [118].

SSA method has been widely used and applied to various optimization problems with great success, but it still suffers from several deficiencies:

1) The method requires prohibitively long relaxation time in order to find solutions with acceptable quality, i.e., SSA consumes too much iteration time due to its Monte Carlo scheme. To guarantee convergence to an exact solution, SSA will require more iterations than complete enumeration for some problems [21].

2) Subtle adjustment of parameters in the annealing schedule, such as the length of the temperature steps during annealing, the temperature range, the number of re-starts and re-direction during the search, etc., is also needed [89, 1, 61, 62, 65].

2.3.2 Chaotic Simulated Annealing

In order to improve the searching ability of the SSA, complex neurodynamics such as chaotic simulated annealing (CSA) is proposed [125]. Compared with the gradient descent dynamics of the HNN models and neural networks with SSA dynamics, neural networks with the CSA have a richer spectrum of dynamic behaviors, such as stable fixed points, periodic oscillations, and chaos. Compared with static neural networks in the past, this kind of neural networks are named as the chaotic neural networks (CNN) due to the totally different neurodynamics.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Nozawa demonstrated the search ability of the chaotic neural networks [80] [124]. Chen and Aihara [125] [21] proposed the chaotic simulated annealing (CSA) by starting with a sufficiently large negative self-coupling in the Aihara-Takabe-Toyoda [7] network when the dynamics is chaotic, and gradually decreasing the self-coupling so that the network eventually stabilizes, thereby obtaining a transiently chaotic neural network (TCNN). Their computer simulations showed that the CSA leads to good solutions for the TSP much more easily compared to the Hopfield-Tank approach [51] [52] and SSA. Chen and Aihara [23] offered the following theoretical explanation for the global searching ability of the chaotic neural network: its attracting set contains all global and local optima of the optimization problem under certain conditions, and since the chaotic attracting set has a fractal structure and covers only a very small fraction of the entire state space, CSA is more efficient in searching for good solutions for optimization problems compared to other global search algorithms such as SSA.

Other kinds of CSA have also been proposed in the previous research. Wang and Smith proposed another chaotic annealing as in [114]. Hayakawa et al. [47] obtained the CSA by adding the chaotic noise into the Hopfield network. He [48] applied the decaying chaotic noise into the Hopfield network and obtained the transiently CSA. Zheng et al. [135] improved the Wang-Smith's chaotic simulated annealing which reaps the benefits of Wang-Smith model and Chen-Aihara model.

There are mainly three significant differences between SSA and CSA [21]:

1) SSA is stochastic on the basis of the Monte Carlo scheme while CSA is deterministic with transiently chaotic dynamics.

2) The convergent processing of SSA is controlled by stochastic "thermal" *fluctuations* while that of CSA is controlled by *bifurcation* structures.

3) SSA fundamentally searches all possible states by temporally changing probability distributions while CSA restricts to a possibly fractal subspace with continuous states. Because the searching region in CSA is usually very small

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

compared with the entire state space, CSA can be expected to perform efficient searching if the restriction is adequate to include a global optimum state or some near-global optimum states.

2.3.3 Mathematical Formulation

The dynamics of CNN with CSA can be summarized as follows. The input-output function (activation function) of each neuron used in the CNN [7] is a typical sigmoidal function:

$$x_i(t) = \frac{1}{1 + e^{-y_i(t)/\varepsilon}} \quad . \tag{2.7}$$

where, x_i is the output of neuron i and y_i is the input of neuron i. ε is the constant controlling the steepness of the sigmoidal or steepness parameter of the activation function with $\varepsilon > 0$. The activation function with different value of ε is shown in Fig. 2.2. It can be seen that the smaller ε will produce the steeper slope curve.

The difference equation of the CNN is:

$$y_i(t+1) = ky_i(t) + \alpha(\sum_{j=1}^{N} w_{ij}x_j(t) + I_i) - z_i x_i(t) \quad . \tag{2.8}$$

where w_{ij} is the connection weight from neuron j to neuron i with $w_{ij} = w_{ji}$ and $w_{ii} = 0$, I_i is the input bias of neuron i, k is the damping factor of nerve membrane ($0 \leq k \leq 1$), α is the positive scaling parameter for inputs, and z_i is the self-feedback connection weight or refractory strength with ($z(t) \geq 0$).

The neuron model of CNN has more complex dynamics in comparison with the conventional neural networks, such as the Hopfield neural networks. The HNN with continuous time or asynchronously discrete-time state transitions, guarantee convergence to a stable equilibrium point but suffer from local minima. On the other hand, the convergence problems of the chaotic dynamics have not been satisfactorily solved although the dynamics of chaotic neural network has an attractive property to move chaotically through fractal structure in the searching space, without getting stuck at local minima. [80] [7]. By taking advantage of

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

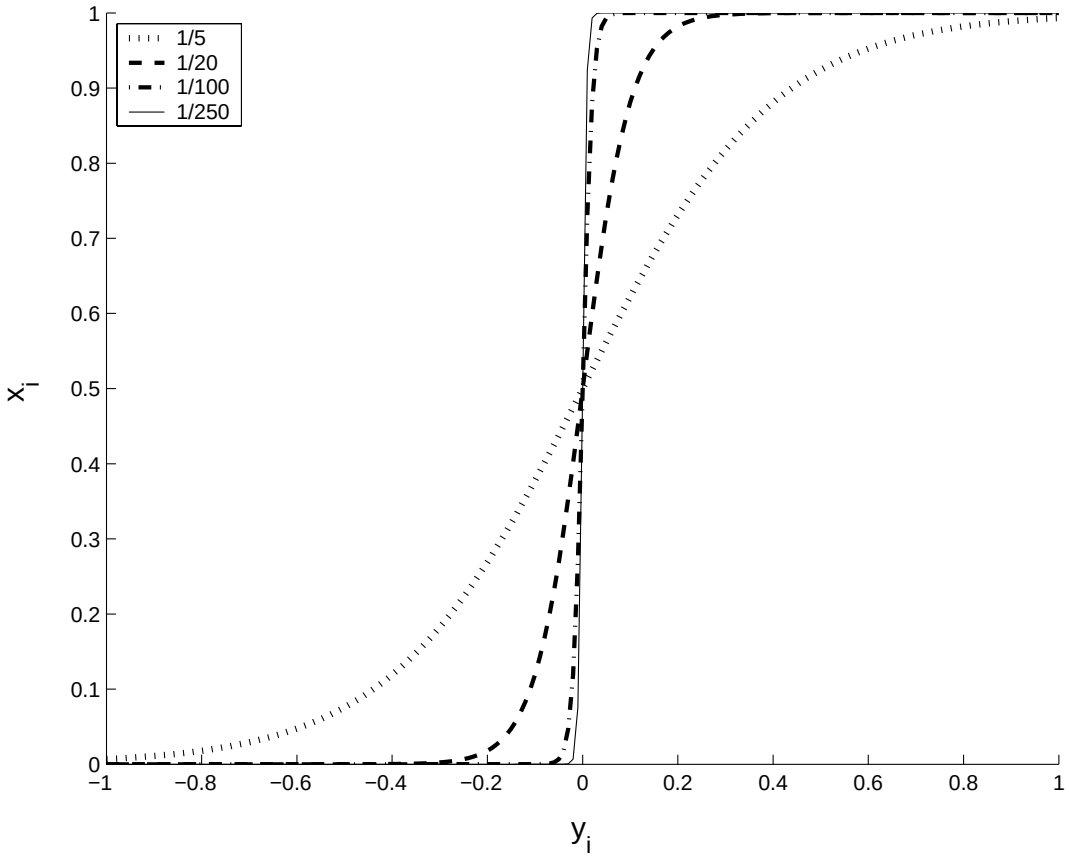


Figure 2.2: The activation function with different values of ε.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

both the convergent dynamics and the chaotic dynamics, Chen and Aihara [125] proposed a transiently chaotic neural network by modifying the CNN in the difference equation:

$$y_i(t+1) = ky_i(t) + \alpha(\sum_{j=1,i\neq j}^{N} w_{ij}x_j(t) + I_i) - z_i(t)(x_i(t) - I_0) \quad . \tag{2.9}$$

where $z_i(t)$ is the self-feedback connection weight or refractory strength ($z(t) \geq 0$), I_0 is the positive parameter.

$$z_i(t+1) = (1-\beta)z_i(t) \quad . \tag{2.10}$$

The updating rule is:

$$\sum_{j=1,i\neq j}^{N} w_{ij}x_j + I_i = -\partial E/\partial x_i \quad \text{input to neuron i} \tag{2.11}$$

where E is the energy function of the neural network.

The difference between CNN and TCNN is the third term on the right-hand sides of eqn. (2.8) and eqn. (2.9), where $z_i x_i(t)$ in eqn.(2.8) is replaced with $z_i(t)(x_i(t) - I_0)$ in eqn. (2.9). This term represents negative (inhibitory) self-feedback or refractoriness with a bias I_0 [7]. It shows that TCNN with transiently chaotic dynamics eventually converges to a stable equilibrium point through successive bifurcations like a route of reversed period-doubling bifurcations. The new variable $z_i(t)$ corresponds to the temperature in usual stochastic annealing process, β governs the bifurcation speed of the transient chaos. Thus, eqn (2.10) denotes an exponential cooling schedule for the annealing [125] [21] [23].

The convergence procedure of the TCNN is fully deterministic, namely, starting from deterministically chaotic dynamics, through, e.g. a reversed period-doubling route with decreasing the value of z which corresponds to the temperature of annealing, finally reaching a stable equilibrium solution [21].

The difference between the TCNN and the HNN is that the a nonlinear term $-z_i(x_i - I_0)$ is added in eqn. (2.9). With the "temperature" $z_i(t)$ tends toward zero,

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

the TCNN eventually reduce to the continuous-time HNN without self-feedback connections [21].

2.4 Noisy Chaotic Neural Network

2.4.1 Stochastic Chaotic Simulated Annealing

Although CSA searches in an efficient manner, CSA has completely deterministic dynamics and is not guaranteed to settle down at a global optimum no matter how slowly the annealing parameter (the neuronal self-coupling) is reduced [109]. Different from the searching direction of SSA that is probabilistically determined by mutual interactions among neurons, CSA is uniquely determined by mutual interactions among neurons. In practical terms, this means that CSA sometimes may not be able to provide a good solution at the end of annealing for some initial conditions of the network, no matter how slowly annealing takes place, i.e., CSA sometimes may not be able to provide a good solution at the conclusion of annealing even after a long time of searching.

By adding decaying stochastic noise into the TCNN, Wang and Tian [118] proposed a new approach to simulated annealing, i.e., stochastic chaotic simulated annealing (SCSA), using a noisy chaotic neural network (NCNN). Compared with CSA, SCSA performs stochastic searching both before and after chaos disappears and is more likely to find optimal or sub-optimal solutions. This novel method has been applied successfully to solving several challenging optimization problems, including the traveling salesman problem (TSP) and the channel assignment problem (CAP) [76] [116]. Furthermore, the NCNN have been applied in image restoration [126] [127] and image denoising [128]. In the application of image denoising, the original Bayesian framework of image denoising is formulated into a constraint optimization problem using continuous relaxing labeling. Compared with the HNN and the TCNN, the results show that the NCNN can offer good quality solutions.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

In [116], the NCNN performed as well as the TCNN in small-size TSPs, such as 10-city and 21-city TSPs. But when used on larger-size TSPs, such as 52-city and 70-city TSPs, the NCNN achieved a better performance compared to the TCNN. In the CAP, the NCNN obtained smaller overall interference (by 2% to 7.3%) compared to the TCNN in all benchmark instances.

2.4.2 Mathematical Formulation

The NCNN model is described as follows [118]:

$$x_{jk}(t) = \frac{1}{1 + e^{-y_{jk}(t)/\varepsilon}} , \tag{2.12}$$

$$y_{jk}(t+1) = k y_{jk}(t) + \alpha \left\{ \sum_{i=1, i\neq j}^{N} \sum_{l=1, l\neq k}^{M} w_{jkil} x_{jk}(t) + I_{ij} \right\} - z(t)(x_{jk}(t) - I_0) + n(t) , \tag{2.13}$$

$$z(t+1) = (1 - \beta_1) z(t) , \tag{2.14}$$

$$n(t+1) = (1 - \beta_2) n(t) , \tag{2.15}$$

where the notations are:

x_{jk} : output of neuron jk ;

y_{jk} : input of neuron jk ;

w_{jkil}: connection weight from neuron jk to neuron il, with $w_{jkil} = w_{iljk}$ and $w_{jkjk} = 0$;

$$\sum_{i=1, i\neq j}^{N} \sum_{l=1, l\neq k}^{M} w_{jkil} x_{jk} + I_{ij} = -\partial E / \partial x_{jk} \tag{2.16}$$

E : energy function which is similar as eqn. (2.5);

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

I_{jk} : input bias of neuron jk ;

k : damping factor of nerve membrane $(0 \leq k \leq 1)$;

α : positive scaling parameter for inputs ;

β_1 : damping factor for neuronal self-coupling $(0 \leq \beta_1 \leq 1)$;

β_2 : damping factor for stochastic noise $(0 \leq \beta_2 \leq 1)$;

$z(t)$: self-feedback connection weight or refractory strength $(z(t) \geq 0)$, $z(0)$ is a constant;

I_0 : positive parameter, which is used as threshold for each neuron, can be a fixed number or variable one;

ε : steepness parameter of the output function $(\varepsilon > 0)$;

$n(t)$: random noise injected into the neurons, in $[-A, A]$ with a uniform distribution;

$A[n]$: amplitude of noise n.

This NCNN model is a general form of chaotic neural networks with transient chaos and decaying noise. In the absence of noise, i.e., $n(t) = 0$, for all t, the NCNN as proposed in eqns. (2.12) - (2.15) reduces to the TCNN in [21]. In order to reveal the search ability of the NCNN, we will examine the nonlinear dynamics of the single neuron model in the next subsection and discuss the selection of model parameters in the model.

2.4.3 Single Neuron Dynamics of the NCNN

The single neuron model for the NCNN is obtained from eqns. (2.12) - (2.15) by letting the number of neurons to be 1. The single neuron model for the TCNN can

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

be obtained with the noise term $n(t)$ set to zero in eqn. (2.18).

$$x(t) = \frac{1}{1 + e^{-y(t)/\varepsilon}} , \tag{2.17}$$

$$y(t+1) = ky(t) + \alpha I_i - z(t)\left[x(t) - I_0\right] + n(t) , \tag{2.18}$$

$$z(t+1) = (1 - \beta_1)z(t) , \tag{2.19}$$

$$A[n(t+1)] = (1 - \beta_2)A[n(t)] . \tag{2.20}$$

where I_i is the input bias of this neuron. We can transform eqn. (2.17) and eqn. (2.18) to a one-dimensional map from $y(t)$ to $y(t+1)$ by substituting eqn. (2.17) into eqn. (2.18).

$$y(t+1) = ky(t) + \alpha I_i - z(t)\left[\frac{1}{1 + e^{-y(t)/\varepsilon}} - I_0\right] + n(t) \tag{2.21}$$

There is a set of parameters in this model, i.e., $k, \alpha, \varepsilon, I_0, z(0), n(0), \beta_1, \beta_2$. It is obvious that different values of these parameters will produce different neurodynamics. In order to investigate the dynamics of the NCNN model, we will vary the parameters above and plot the neuron dynamics and choose the set of parameter values which can produce richer and more flexible dynamics. The parameters α, β_1, ϵ and I_0 have been discussed in detail in [21].

In [21], L. Chen and K. Aihara found that α represents the influence of the energy function on the neuro-dynamics. It is the balance between the self-feedback term $-z(t)\left[x(t) - I_0\right]$ and the gradient term $-\partial E/\partial x_{jk}$. The effect of the self-feedback term is to generate the chaotic dynamics, and the gradient term is used to create the convergent dynamics. So if α is too large, the influence of the energy function will become too strong to generate the chaos. On the contrary, if α is too small, the energy function will not be able to sufficiently reflected in the neuro-dynamics. They also noticed that the parameter β_1 can be considered as a damping speed parameter of the megatove self-feedback strength, which controls the annealing process. Hence in order to produce stronger and longer chaotic chaotic dynamics, the value of β_1 should be set smaller. In their simulation on 10 city TSP problem [21], β_1 is set as small as 0.005, or even 0.003.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

As the NCNN is an extension model of the TCNN in [21], we will use the same rules as in the TCNN when choosing parameters. The values of these parameters used in our simulations are listed as below:

$$\alpha = 0.015; \beta_1 = 0.001; \varepsilon = 1/250; I_0 = 0.65. \tag{2.22}$$

Where α, β_1, ϵ will be kept unchanged in all the simulated optimization problems. The value of I_0 will also be the same in all our methods excepts the NCNN-VT, where the parameter I_0 will be chosen as different values in order to achieve the variable thresholds.

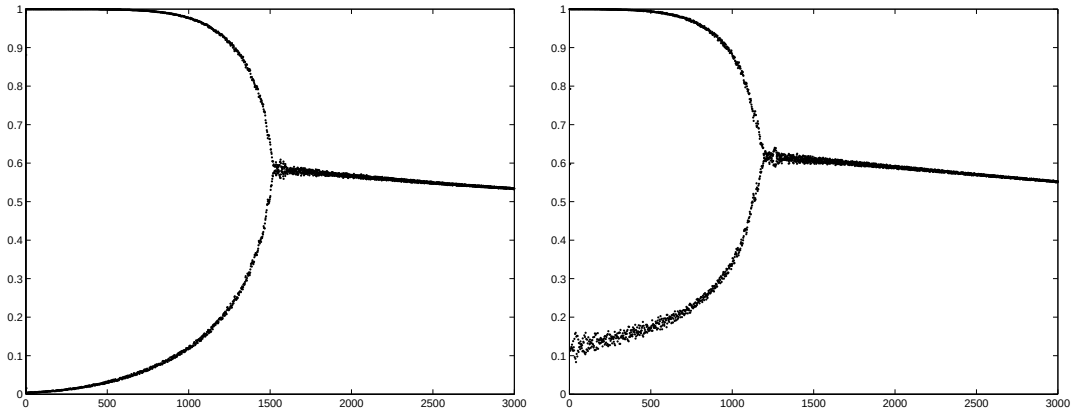
Since β_2 is the damping factor of noise which has the similar effect as the chaos damping factor, β_1, we choose $\beta_2 = 0.0001$. We will then vary only the parameters left, i.e., the nerve membrane damping factor k, the initial negative self-interaction $z(0)$, and the initial noise amplitude $n(0)$, to investigate the dynamics, while keeping the other parameters fixed.

Fig. 2.3 shows the dynamics of the single neuron for different parameter k. The x-axis of each sub-figure is time step t, and the y-axis is the output of the neuron $x(t)$. We see from Fig. 2.3 that the larger the value of k, the more the bifurcations. Since chaos is important for searching and k is between 0 and 1, we choose $k = 0.9$ as showed in Fig. 2.3 (d).

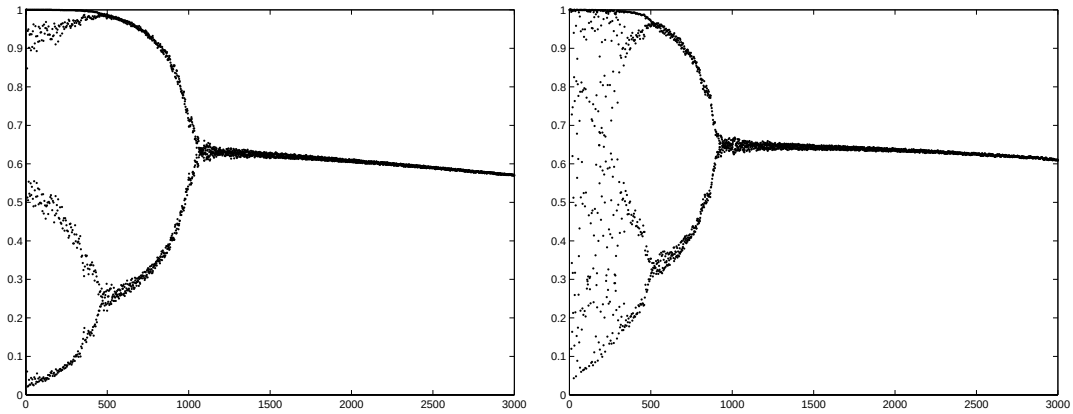
Fig. 2.4 shows that a small value of $z(0)$, e.g., 0.01, cannot produce chaos. Larger values of $z(0)$ lead to more bifurcations as showed in Fig. 2.4 (c) and (d). But larger $z(0)$ also lead to more iteration steps until the chaos disappears, and a solution is found which inevitably results in longer computational time. Hence we choose a tradeoff between efficiency and solution quality, i.e., we use the value of $z(0) = 0.08$ in this paper.

Fig. 2.5 reveals that too much additive noise destroys the bifurcations (Fig.4 (d)). On the other hand, if the magnitude of the additive noise is too small, it does not take effect on the stochastic search. Thus in this paper, we choose the initial noise amplitude as $n(0) = 0.001$.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



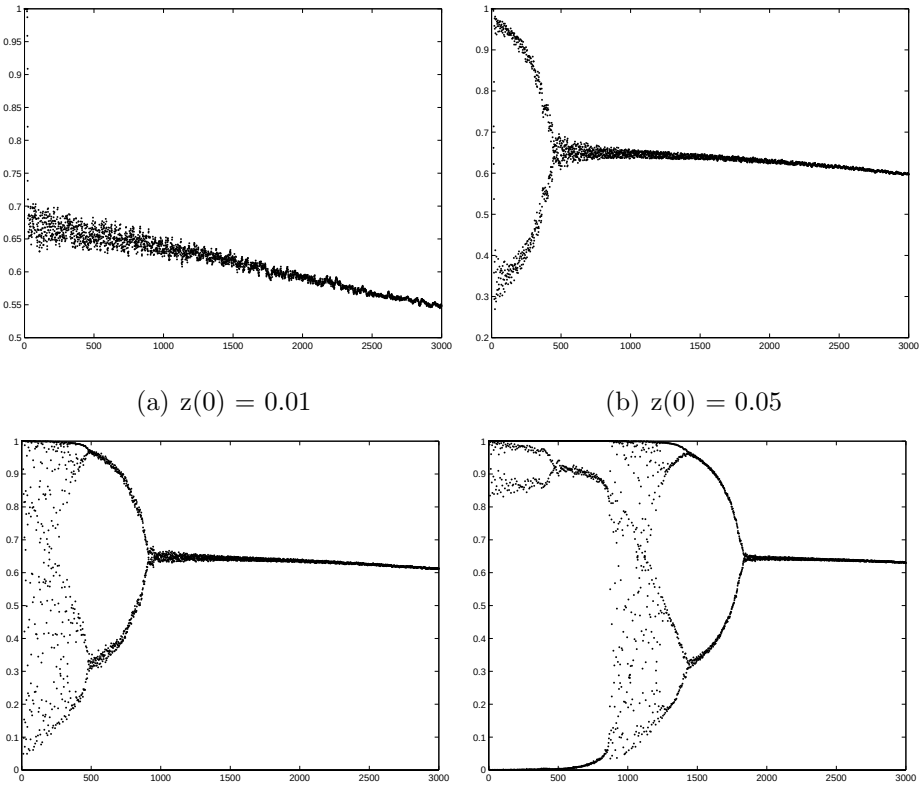
(a) k = 0.1 (b) k = 0.5



(c) k = 0.7 (d) k = 0.9

Figure 2.3: The neuro-dynamics in the single neuron model with different parameter k. $\alpha = 0.015$, $\beta_1 = 0.001$, $\beta_2 = 0.0001$, $\varepsilon = 1/250$, $I_0 = 0.65$, $z(0) = 0.08$, and $n(0) = 0.001$.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

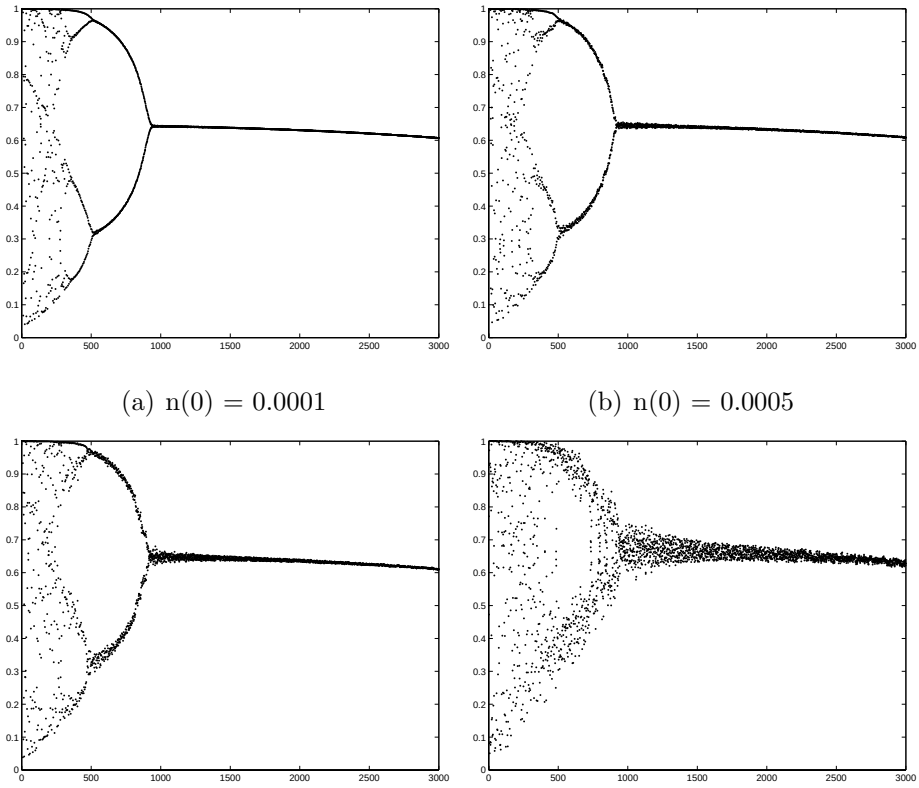


(a) z(0) = 0.01 (b) z(0) = 0.05

(c) z(0) = 0.08 (d) z(0) = 0.2

Figure 2.4: The neurodynamics in the single neuron model with different parameter $z(0)$. $k = 0.9$, $\alpha = 0.015$, $\beta_1 = 0.001$, $\beta_2 = 0.0001$, $\varepsilon = 1/250$, $I_0 = 0.65$, and $n(0) = 0.001$.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(a) n(0) = 0.0001 (b) n(0) = 0.0005

(c) n(0) = 0.001 (d) n(0) = 0.005

Figure 2.5: The neurodynamics in the single neuron model with different parameter $n(0)$. $k = 0.9$, $\alpha = 0.015$, $\beta_1 = 0.001$, $\beta_2 = 0.0001$, $\varepsilon = 1/250$, $I_0 = 0.65$, and $z(0) = 0.08$.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Based on the above discussions on the selection of model parameters for the NCNN, we finally choose the set of parameters as in eqn. (2.23).

$$k = 0.9; \varepsilon = 1/250; I_0 = 0.65; z(0) = 0.08;$$
$$n(0) = 0.001; \beta_1 = 0.001; \beta_2 = 0.0001. \tag{2.23}$$

The difference between the TCNN and the NCNN is the stochastic nature of the NCNN which the TCNN lacks. Chaos disappears after around 950 iterations through the reverse period-doubling bifurcations in the TCNN, whereas in the NCNN shown in Fig. 2.5 (c), after the chaos disappears, the additive noise still remains and decays with time. With both stochastic nature for global searching and chaotic characteristic for efficient searching, the NCNN gradually approaches a dynamic structure similar to that of the Hopfield neural network and converges to a stable fixed point.

2.4.4 Stability Analysis for the NCNN

The stability of the NCNN is related to the analysis of the HNN and the TCNN, because the NCNN is a general neural network model. In [22], Chen and Aihara proved the asymptotical stability of the TCNN under several conditions. Because the TCNN has the same formulation as the NCNN, the theoretical results on the TCNN hold for the NCNN.

In this thesis, we start to investigate the stability of the NCNN from the perspective of the network energy. Kwok and Smith [68, 69] introduce a unified framework to investigate the different neural network models by adding a additional term in the Hopfield neural network energy. They proposed the unified energy framework which includes Hopfield energy part and additional energy part. We rewrite the energy function of the HNN as below:

$$E_{Hop} = -\frac{1}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} w_{ij}x_i x_j - \sum_{i=0}^{N-1} I_i x_i + \frac{1}{\tau}\sum_{i=0}^{N-1}\int_0^{x_i} g^{-1}(\eta)d\eta$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

where $\mathbf{w} = \{w_{ij}\}$ is the weight matrix, I_i is the firing threshold of neuron i, τ is a positive constant and g is a continuous activation function as in Fig. 2.2. y_i is the internal state of neuron i, and x_i is the output of neuron i with $x_i(t) = f(y_i(t))$.

As discussed in section 2.2.2, the HNN converges to a minimum by using the updating rule given by:

$$\frac{dy_i}{dt} = -\frac{\partial E}{\partial x_i} \tag{2.24}$$

The general form of the computational energy E in [68] is described as follows:

$$\begin{aligned} E = &-\frac{1}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} w_{ij}x_i x_j - \sum_{i=0}^{N-1} I_i x_i \\ &+\frac{1}{\tau}\sum_{i=0}^{N-1}\int_0^{x_i} g^{-1}(\eta)d\eta + H(x_i, w_{ij}, I_i)\ . \end{aligned} \tag{2.25}$$

or simply $E = E_{Hop} + H$, where the H is the additional term which intend to include the chaotic dynamics.

In order to study the stability of the NCNN, we will firstly propose the general form of the energy E_{ncnn} and investigate the stability based on this energy model. In this context, we follow the method in [68] and proposed the energy function of the NCNN model as:

$$H_{ncnn} = \frac{\lambda(t)}{2}\sum_{i=1}^{N} x_i(x_i - l - p)\ . \tag{2.26}$$

where $\lambda(t)$ is a positive decay parameter which monotonously decreases with time. l and p are positive parameters.

The energy of the NCNN model is then the summation of the E_{Hop} and H, i.e.,

$$\begin{aligned} E_{ncnn} = &-\frac{1}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} w_{ij}x_i x_j - \sum_{i=0}^{N-1} I_i x_i + \frac{1}{\tau}\sum_{i=0}^{N-1}\int_0^{x_i} g^{-1}(\eta)d\eta \\ &+\frac{\lambda(t)}{2}\sum_{i=1}^{N} x_i(x_i - l - p)\ . \end{aligned} \tag{2.27}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

By applying the updating rule in eqn. (2.24), we have

$$\frac{dy_i}{dt} = 1 - \frac{y_i}{\tau} + \sum_j^N w_{ij}x_j + I_i - \lambda(x_i - \frac{l}{2} - \frac{p}{2}) \ . \tag{2.28}$$

Using the Euler discretization

$$y_i(t+1) = (1 - \frac{\Delta t}{\tau})y_i(t) + \Delta t \left(\sum_j^N w_{ij}x_j(t) + I_i \right) - \Delta t\lambda(t)(x_i - \frac{l}{2}) + \Delta t\frac{p}{2}\lambda(t) \ . \tag{2.29}$$

Compared with the difference equation of the NCNN in eqn. (2.13), the following relationships hold:

$$1 - \frac{\Delta t}{\tau} = k, \quad \Delta t = \alpha \quad \Delta t\lambda(t) = z(t), \quad I_0 = \frac{l}{2}, \quad \Delta t\frac{p}{2}\lambda(t) = n(t) \ . \tag{2.30}$$

From the eqn. (2.30), we can determine the values of parameters $\lambda(t)$, l, and p in eqn. (2.26).

After the formulation of the energy function of the NCNN, we now investigate the stability of the NCNN. The time derivative of the energy E in eqn. (2.27) is:

$$\begin{aligned} \frac{dE_{ncnn}}{dt} &= \frac{dE_{Hop}}{dt} + \frac{dH_{ncnn}}{dt} \\ &= -\sum_{i=1}^N \frac{dx_i}{dt}(\sum_{j=1}^N (w_{ij}x_j + I_i - \frac{y_i}{\tau}) + \frac{\lambda(t)}{2}(2x_i - l - p)\frac{dx_i}{dt} \\ &= -\frac{dx_i}{dt}\frac{dy_i}{dt} + \frac{dx_i}{dt}\lambda(t)\left[x_i - \frac{l+p}{2}\right] \\ &= -(\frac{dx_i}{dt})^2(g^{-1}(t))' + \frac{dx_i}{dt}\lambda(t)\left[x_i - \frac{l+p}{2}\right] \end{aligned} \tag{2.31}$$

where $g^{-1}(t)$ is the inverse function of the activation function which is increasing, thus the first part of the eqn. (2.31) is smaller than 0. Since $\lambda(t)$ is a decaying term which is similar with $z(t)$, i.e., $\lambda(t+1) = (1-\beta_3)\lambda(t)$ with $\beta_3 > 0$, the second term will disappear after a certain number of iterations.

From the energy perspective, the NCNN model will not monotonously decrease as time evolves. On the contrary, the chaotic dynamics makes the energy

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

fluctuates in the early stages. When the chaos disappears, the second term in eqn. (2.31) will also disappear, then the energy will perform like a HNN and decrease with time step t, and finally converge to the nearest optimal solution.

2.4.5 Computational Complexity of the NCNN

On the issue of computational complexity of neural networks, there are some research work on the discrete neural networks [2] [129] [83] [81] [104] [103] . Orponen surveyed some of the central results in the complexity theory of discrete neural networks [81] and showed that in a symmetric simple network of N neurons with integer weights w_{ij} in the HNN, the stabilization requires at most a total of

$$3\sum_{j<i}|w_{ij}| = O(n^2 \cdot max_{i,j}|w_{ij}|) \ .$$

neuron state changes, under an asynchronous update rule [35]. Under synchronous update rule, a similar bound holds [44].

For a noisy chaotic neural network with $N \times M$ neurons, it is difficult to determine or estimate the exact iterations needed for the network to stabilize. However, it can be seen that each neuron was updated once in asynchronously mode and the computational complexity for the neural network in one iteration is $O(NM)$ or $O(N^2)$ if $N = M$.

2.5 Concluding Remarks

In this chapter, we have reviewed the mathematical formulation of the HNN and the stability of the HNN. Although the feasibility of the HNN in solving the TSP has been doubted, many explanations for the poor solution quality of TSP have been made in terms of energy function formulation and parameter selection. Because Hopfield neural networks are intrinsically parallel and easy for rapid hardware

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

implementation, HNN and modified versions of the HNN have been widely used in solving different optimization problems.

Since the original Hopfield neural network (HNN) can be easily trapped in local minima, stochastic simulated annealing has been combined with the HNN, i.e., mean field annealing (MFA). SSA is known to relax to a global minimum with probability 1 if the annealing takes place sufficiently slowly, i.e., at least inversely proportional to the logarithm of time. In a practical term, this means that SSA is capable of producing good (optimal or near-optimal) solutions for many applications, if the annealing parameter (temperature) is reduced exponentially with a reasonably small exponent.

So far, a number of artificial neural networks with chaotic dynamics have been investigated toward more complex neuro-dynamics. Among them, CNN is proposed by considering graded responses, relative refractoriness and spatio-temporal summation of inputs [7]. In spite of its simple equations, CNN contains various coexisting attractors, not only of fixed points and periodic points but also of strange attractors. In order to tackle the convergence problems of the chaotic dynamics, Chen and Aihara [125] proposed the TCNN by reducing the self-feedback or refractoriness of each neuron. The TCNN shows the ability in solving the TSP without the convergence problem.

By adding the decaying noise into the TCNN model, Wang and Tian [118] proposed the NCNN which has transient chaos and stochastic noise. Because the TCNN with CSA is a fully deterministic dynamics, the stochastic nature of noise helps the neural networks to search more efficiently compared with the CSA. Through the applications of the NCNN on the NP-hard COPs including TSP and CAP and in the area of image restoration and image denoising, the NCNN outperforms the HNN and TCNN in computation time and solution quality.

We have discussed the selection of parameters in the NCNN model by investigating the neurodynamics of the single neuron. The parameters includes the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

model parameters in the NCNN model and weighting coefficients in the energy function. The rules of selection of model parameters is that the set of parameters will produce more flexible and richer dynamics. As we will see in the applications to the COPs in the later chapters, these model parameters will vary only in very small range and some parameters are fixed in different instances with increasing problem size.

As the NCNN is a general form of neural network model, the stability if the HNN can be seen as the special case of the NCNN. We have analyzed the stability of the NCNN by adopting the unified energy framework. The NCNN performs in a chaotic phase in the early stage and the energy of the neural network do not monotonously decrease. After the chaos disappears, the NCNN behaves like the HNN and the energy will decrease with time. Then the neural network will stabilize and the NCNN will converge to the nearest optimal or sub-optimal solutions.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 3

Gradual Noisy Chaotic Neural Network and Noisy Chaotic Neural Network with Variable Threshold

3.1 Introduction

The successful applications of the NCNN to various NP-hard COPs have proven the ability of the NCNN model. However, the utilization of the NCNN met some difficulties. First of all, there are some COPs whose objectives are hard to formulate the energy function in neural networks (as can be seen in the application of the BSP where it is hard to formulate the first objective of the problem into the energy function of the neural network). Another situation is that the choice of the weighting coefficients in energy function are often obtained through trial and error, which is hard and tedious when the energy function consists of many terms including objective terms and constraint terms.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

We will propose two types of original neural networks by combining two schemes with the NCNN model [1]. The two types of neural networks proposed in this thesis are the gradual noisy chaotic neural network (G-NCNN) and the noisy chaotic neural network with variable threshold (NCNN-VT). The G-NCNN is the hybridization of the gradual expansion scheme (GES) with the NCNN, while the NCNN-VT is the combination of adaptive mapping scheme (AMS) with the NCNN. Both the two proposed model maps the objective function of the COPs into the scheme while the constraints terms of the COPs are formulated in the energy functions. Through the separation of objectives from the constraints, the selection of the weighting coefficients of the energy function can be simplified, as seen from the applications and result discussions in the later chapters.

In this chapter, we will firstly discuss the GES in Section 3.2 and propose detailed implementation of the G-NCNN. The difference between the G-NCNN and the NCNN will be discussed in the end of the section. In Section 3.3, we will propose the AMS and the model definition of the NCNN-VT. The originality of the proposed NCNN-VT will be introduced and the difference between the NCNN-VT and the NCNN will be investigated in detail.

3.2 Gradual Noisy Chaotic Neural Network

3.2.1 Gradual Expansion Scheme

Funabiki et al. proposed a novel neural network by combining the gradual expansion scheme (GES) with binary neural network [36]. They called it the gradual neural network (GNN). In this work, we adopt the GES and combine it with our NCNN model. In order to prevent the local minima problem, the GNN adopts many heuristics including hill-climbing and problem-specific functions [36]. In the

[1] The two schemes can also applied to the TCNN. Due to the fact that the NCNN outperforms the TCNN, we only applied these two schemes on the NCNN model

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

aspect of dynamics, the G-NCNN has richer and more flexible neurodynamics which can maintain a more effective search than the GNN in the solution space.

The essence of the gradual expansion scheme is that the number of neurons in the neural networks are gradually increased. Usually, the number of neurons the neural networks needed for solving a COP is determined by the problem, e.g., a N-city TSP problem, $N \times N$ neurons needed to compose a solution space. For the NCNN, the N^2 neurons are needed at the start of the neuron computation. But if using the gradual expansion scheme, the neural network needs only partial number of neurons at the beginning of neuron computation. The neurons left are gradually added to the existing group of neurons when needed.

Now we consider the reason why we need the GES and its advantages. There is one common kind of objective in the COPs which needs to find the minimal "cost", where the cost can have many different meanings in different problems, e.g., minimal interference, minimal delay, or minimal length of path. The benefit of the GES is that the objective of the COP can be realized during the stage of the gradual expansion. In order to achieve the objective that the solution found by the neural networks is with the minimal cost, neurons are divided into several groups and activated or added in several stages with each stage only adopt one group of neurons with smallest cost left over. Through the GES, we do not need to formulate the objective ("minimal cost") in the energy function because we think that the objective (to find solution minimal cost) is realized in the gradual expansion stages which let the neurons with smaller cost be included in neuron computation in earlier stage. When dealing with this kind of objective, we can adopt the GES naturally through the following steps:

1) Assume there are $N \times M$[1] neurons needed for the problem. Compute the cost matrix $C[N \times M]$, i.e., the cost of neuron ij if neuron ij is selected in the final solution.

[1] N and M can have the same value.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

2) Sort the neuron in ascending order of the cost.

3) Divide the $N \times M$ neurons into P groups $(G_1, G_2, ..., G_P)$, with each group have p neurons, where p is the maximum integer less than or equal to NM/P. Group G_1 is the group of neurons with smallest cost and other groups G_2 to G_P contain the neurons with larger cost with an ascending order of cost.

4) Add the neurons in the first group G_1 to the neural network and apply the neural network computation. If the feasible solution is found then exit, the found solution is the final solution. If the pre-defined steps are met and still no solution found, then apply the gradual expansion scheme by adding the neurons in the G_2 and apply the neural network computation again.

Fig. 3.1 shows one example of the G-NCNN with 24 ($N = 6$ and $M = 4$) neurons. The black squares indicate the newly added neurons in the stage and the dark gray squares are consecutively activated neurons in the past stages. The parameter P is 4 and each group has exactly ($NM/P = 24/4 = 6$) neurons. Fig. 3.1 (a) is the cost matrix C for this example. Each item c_{ij} in cost matrix stands for the cost value for the selection of the neuron ij in the final solution. At the beginning of the neural computation, only 1 group with minimal cost are activated. In the stage 2, another group of neurons are added and so forth until all neurons are included in the neural network computation.

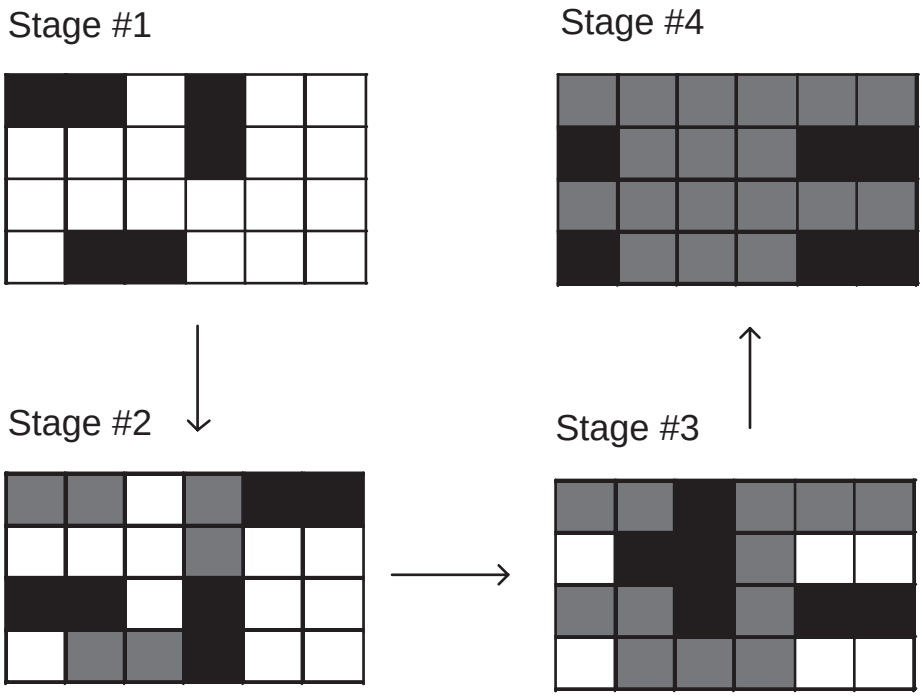
3.2.2 Discussions on the G-NCNN

The G-NCNN is the hybridization of the GES with the NCNN. Compared with the common NCNN model, G-NCNN offers a new way to formulate the objective of the problems. By allowing the neurons expanded gradually in several stages according to the ascending cost order, the neural network starts to seek the solution in partial number of neurons with smaller costs compared with other un-activated neurons. By this way, the objective of the problem which aims to find solution with minimal

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

10	10	40	1	23	22
50	29	28	19	60	100
30	30	45	25	35	34
48	4	10	26	50	100

(a)



(b)

Figure 3.1: An example of the gradual expansion scheme in the G-NCNN. (a) The cost. (b) The multi-stage activation of the G-NCNN by adopting the gradual expansion scheme. The black squares indicate the newly added neurons in the stage and the dark gray squares are consecutively activated neurons in the past stages.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

cost can be realized. The advantage of this G-NCNN model is that the formulation of problem in energy function is easier and the selection of weighting coefficients will be simpler. However, the multi-stage search entangled by the G-NCNN may lead to the disadvantage that longer computation time maybe needed than normal neural networks if the feasible solution is found only in the last stage.

3.3 Noisy Chaotic Neural Network with Variable Threshold

The previous methods including HNN, TCNN and NCNN all combine the constraint satisfaction and the objective optimization in one energy function, which result in the poor performance in convergence if the parameters in energy function are not tuned well. Moreover, the balance between the constraint term and optimization term is a delicate work which needs tedious selections through trial and error method. By separating the optimization term from the constraint ones, we propose the noisy chaotic neural network with variable threshold which handles the constraints in the energy function while deals with the optimization term in the selection of neuron firing.

3.3.1 Adaptive Mapping Scheme

Adaptive Mapping Scheme (AMS) aims to map the objective of optimization problem into the probability of firing of each neuron. If the objective of the problem is to find the solution with minimal cost, then the neurons with smaller cost will have larger probability to be selected in the final solution matrix. By investigating the single neuron dynamics of the NCNN model, it can be seen that the positive parameter I_0 in the NCNN model is responsible for the neuron firing probability. The single neuron dynamics of the NCNN by varying the parameter I_0 is shown in Figs. 3.2 to 3.5 (the x-axis is the time steps t, the y-axis is the output of neuron

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

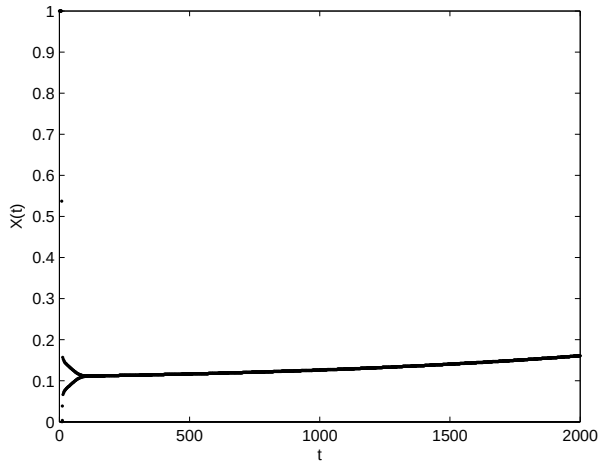


Figure 3.2: The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.1$.

$x_{ij}(t)$). We can see clearly that if the value of parameter I_0 is 0.3, the output of neuron is very close to 0.3 after the network passes the last bifurcation-2 in Fig. 3.3. After the bifurcation point, the neural network slowly converge to a stable point at about 0.5. The same pattern can be observed in other figures. It means that different value of I_0 ($0 < I_0 < 1$) induces different probability of neuron firing. The neuron with large value of I_0 will be selected to be firing (neuron output equals 1) with bigger probability, whereas, the small value of I_0 will result in less chance the neuron to be firing.

3.3.2 Model Definition

The difference between the NCNN and the NCNN-VT model is that the positive parameter I_0 in eqn. (2.13) of the NCNN becomes a variant labeled as I_{jk}, where

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

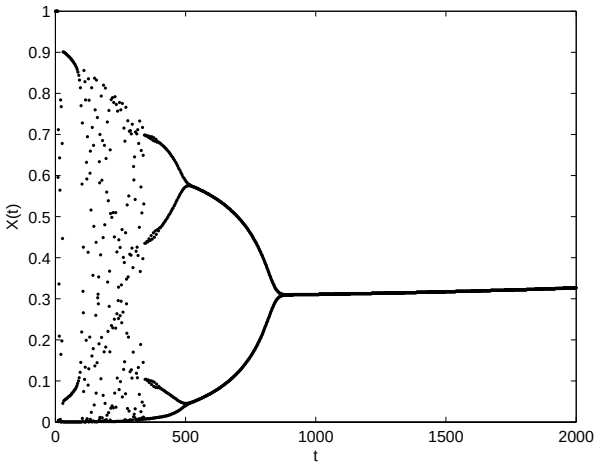


Figure 3.3: The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.3$.

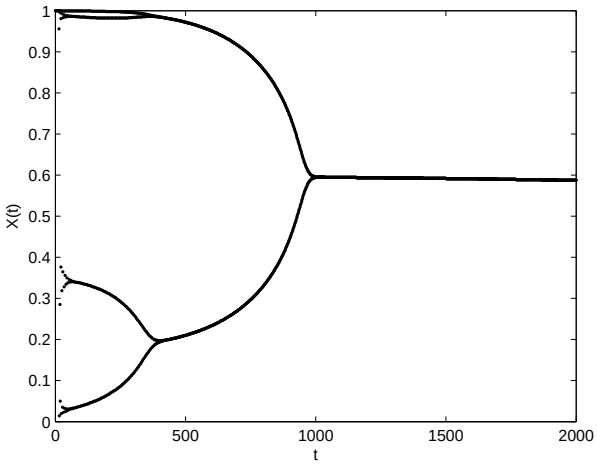


Figure 3.4: The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.6$.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

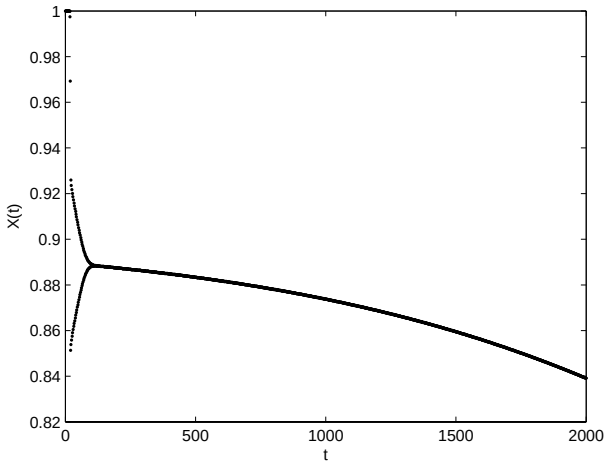


Figure 3.5: The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.9$.

jk stands for the neuron jk, i.e.,

$$y_{jk}(t+1) = k y_{jk}(t) + \alpha \left\{ \sum_{i=1, i\neq j}^{N} \sum_{l=1, l\neq k}^{M} w_{jkil} x_{jk}(t) + I_{ij} \right\}$$

$$-z(t)\left[x_{jk}(t) - I_{jk}\right] + n(t)\,. \tag{3.1}$$

where I_{jk} is not a constant parameter but a variable which determines the selection of firing of each neuron. The value of I_{jk} is related to the problem optimization term.

3.3.3 Mapping Functions

The mapping function connects the problem's cost with the parameter I_{ij}.

$$I_{jk} = f(c_{jk}), \quad j = 1, 2, ..., N; k = 1, 2, ..., M \tag{3.2}$$

The mapping function can be a linear or nonlinear function which transform the cost into the probability value between 0 and 1. Normally, a linear mapping

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

is adopted:

$$f(c_{jk}) = 1 - \frac{c_{jk} - c_{min}}{c_{max} - c_{min}} \,. \tag{3.3}$$

where c_{jk} is the jk element in the cost matrix C, c_{max} is the maximum value among the cost matrix and c_{min} is the minimum value in the cost matrix. Other forms of nonlinear functions can be adopted, e.g., the quadratic functions.

Fig. 3.6 shows the various mapping from the cost into the firing probability for each neuron in the neuron matrix. Fig. 3.6 (a) is the mapping function in eqn. (3.4). Fig. 3.6 (b) and (c) are nonlinear functions. In this thesis, we choose a simple quadratic function for Fig. 3.6 (b) as:

$$I_{ij} = Ad_j^2 + Bd_j + C \,. \tag{3.4}$$

where A and B are coefficients of nonlinear mapping function and can be deduced from constant d_{min} and d_{max}. For the quadratic function in eqn. (3.4) has two points $(d_{min}, 1)$ and $(d_{max}, 0)$, we have:

$$-\frac{B}{2A} = d_{max} \,.$$
$$Ad_{min}^2 + Bd_{min} + C = 1 \,.$$
$$Ad_{max}^2 + Bd_{max} + C = 0 \,.$$

From the above two equations, it can be computed that:

$$A = \frac{1}{(d_{max} - d_{min})^2} \,.$$
$$B = -\frac{2d_{max}}{(d_{max} - d_{min})^2} \,.$$
$$C = \frac{d_{max}^2}{(d_{max} - d_{min})^2} \,.$$

Then the nonlinear mapping function for Fig. 3.6 (b) can be expressed as:

$$I_{ij} = \frac{1}{(d_{max} - d_{min})^2} d_j^2 - -\frac{2d_{max}}{(d_{max} - d_{min})^2} d_j + \frac{d_{max}^2}{(d_{max} - d_{min})^2}$$
$$= (\frac{d_{max} - d_j}{d_{max} - d_{min}})^2 \,. \tag{3.5}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

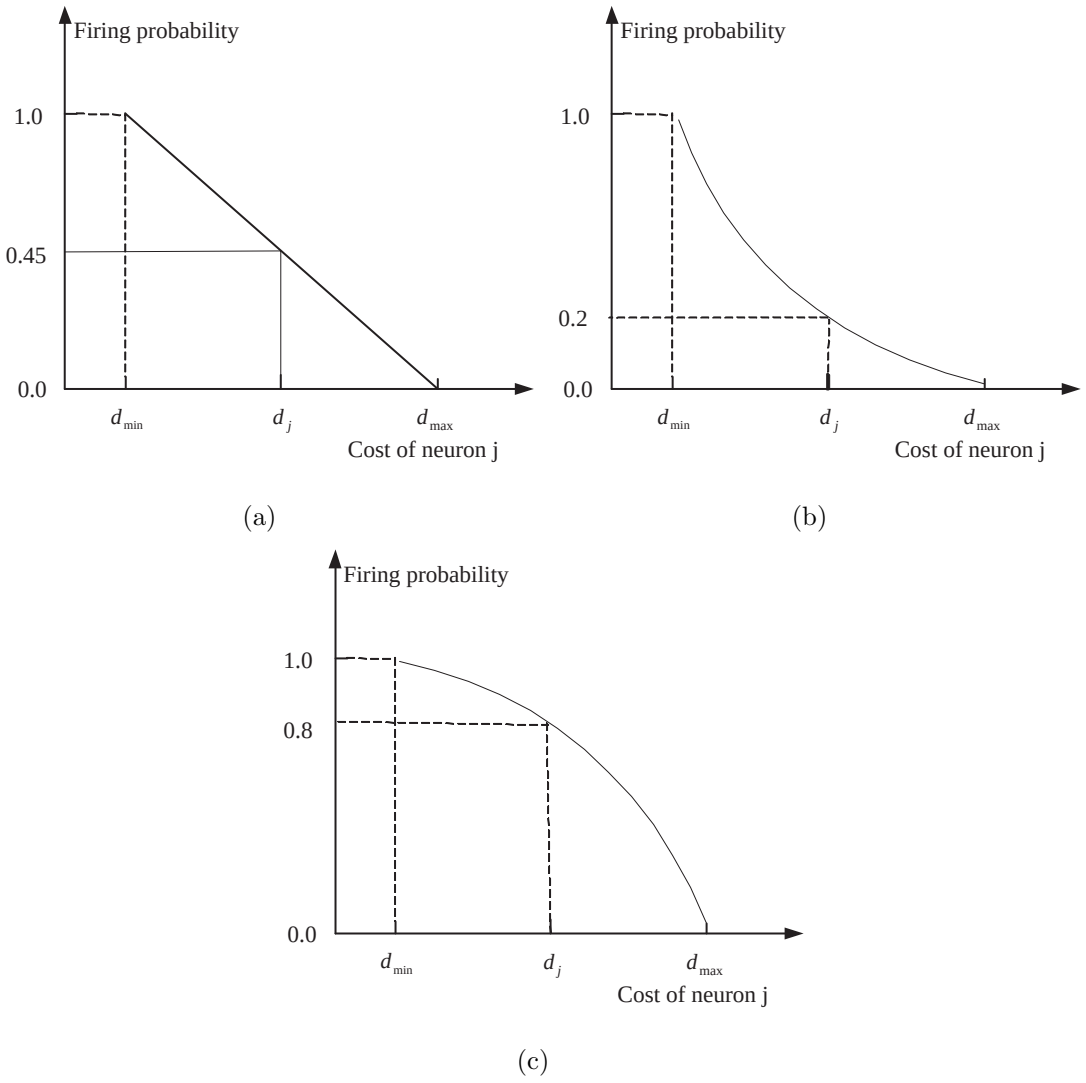


Figure 3.6: Different mapping functions for the NCNN-VT. (a) a linear mapping function. (b) a nonlinear mapping function, case 1. (c) a nonlinear mapping function, case 2.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

For the function in Fig. 3.6 (c), we choose:

$$d_{ij} = Dd_j^2 + Ed_j + F \,. \tag{3.6}$$

For the quadratic function in eqn. (3.6) has two points $(d_{min}, 1)$ and $(d_{max}, 0)$, we have:

$$-\frac{E}{2D} = 0 \,.$$
$$D + F = d_{min} \,.$$
$$F = d_{max} \,.$$

From the above two equations, it can be computed that:

$$D = d_{min} - d_{max} \,.$$
$$E = 0 \,.$$
$$F = d_{max} \,.$$

Then the nonlinear mapping function for Fig. 3.6 (c) can be expressed as:

$$I_{ij} = \sqrt{\frac{d_{max} - d_j}{d_{max} - d_{min}}} \,. \tag{3.7}$$

From the mapping functions in eqn. (3.3), eqn. (3.5), eqn. (3.7), and Fig. 3.6, it can be seen that the neuron ij with smaller cost d_{ij} will have larger or higher probability to fire while the ones with larger cost will be inhibited to fire as shown in Fig. 3.6. The performance of the adaptive mapping function and the NCNN-VT will be demonstrated in the application to the frequency assignment problem in Chapter 6.

3.4 Parameter Discussion

Although the proposed neural network models have many parameters, the rules of parameter selection as discussed in Section 2.4.3 are applied here. Furthermore,

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

these parameters will not vary with simulation problems except the parameters β_1, β_2, and the noise level $A[n(0)]$.

The parameter β_1 is used to control the damping speed of the chaos and β_2 is employed to take care of the noise. These two parameters will be chosen according to different simulation problems in order to produce more flexible nerodynamics. The parameter noise level $A[n(0)]$ is the initial noise amplitude which determines how much noise is injected into the chaotic neurodynamics. We choose the parameter $A[n(0)]$ in order to obtain the appropriate noisy dynamics, which will produce enough stochastic wandering while not destroy the chaotic neurodynamics. In later chapters, the values of these parameters will be listed in details for each simulation problem.

3.5 Concluding Remarks

In this chapter, we have proposed the two extended neural network model based on the NCNN. The motivation of the proposal of new models is that the NCNN faced the difficulties in parameter tuning and problem formulation using energy function. By adopting the two new schemes and combining them with the NCNN, we obtained the G-NCNN and the NCNN-VT, which separate the objective of the problem from the constraints. Thus it makes the problem formulation easier and the selection of network parameters simpler.

The G-NCNN is a model that combines the gradual expansion scheme with the NCNN. The difference between the G-NCNN and the NCNN is that the network structure is not fixed, i.e., the number of neurons needed in the network is gradually expanded. By allowing the neurons with smallest cost be computed first, the G-NCNN can achieve the objective that the solution found by the model is the optimal one with minimal cost.

We have proposed another neural network by adopting the adaptive mapping

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

scheme with the NCNN, which is a NCNN-like model by varying one parameter in the original NCNN. By investigating the neurodynamics of the NCNN, we found that the parameter I_0 in the NCNN model determines the firing probability of the neuron. Large value of I_0 ($0 < I_0 < 1$) indicates the large probability that the neuron output will be 1, i.e., according to the various definitions of the neuron output in different combinatorial optimization problem, that means the path will be selected, the network link will be chosen, the decision will be made, or the assignment will be allocated.

In the following chapters, we will discuss the applications of the NCNN, G-NCNN, and NCNN-VT on three different NP-hard COPs occurred in communications networks. The simulation results will reveal the performance of the proposed network models compared with previous methods.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 4

Solving the Topological Optimization Problem Using the Noisy Chaotic Neural Network

4.1 Introduction

The topological optimization problem (TOP) of networks has its effects in a wide spectrum of areas including telecommunications, computer networks, electricity distribution, gas pipelines, and sewer networks [32]. In this kind of networks, the design problem in the network concerns the best way to layout all the network components in order to minimize the total costs while meeting a certain performance criteria, i.e., finds a topology of networks which minimize the number of connections of nodes (links), while meets the requirements of connectivity. Usually these requirements or criteria include parameters such as transmission delay, throughput, or the reliability of network [56]. In this chapter, My focus is on this topology network design problem with all-terminal reliability (ATR) constraint, i.e. topology optimization problem (TOP in short). The topological optimization of networks with ATR has major importance in the computer communication

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

industry because of its wide applications.

In a communication network, the all-terminal network reliability (also called uniform or overall network reliability) is defined as the probability that every pair of nodes can communicate with each other [27], i.e., an acceptable probability that every pair of nodes can communicate with each other [56]. This is distinguished from the more common source-sink (s-t) network where reliability is measured from the probability that the source is connected with the sink. AboElFotoh *et al.* [4] discussed the calculation of the 2-terminal reliability in radio network. And in [5], they provided the discussions on the reliability problem in wireless distributed sensor networks. In the context of graph theory, if the network is formulated as a graph, the ATR means that the network forms at least a spanning tree [29].

The network design problem has two different forms. One is to choose a set of links for a given set of nodes to maximize reliability given a cost constraint, the other is to choose a set of links for a given set of nodes to minimize cost given a minimum network reliability constraint. In this chapter, I aim to solve the latter, i.e., design of networks considering all-terminal reliability constraint with minimum cost. This topological optimization problem is an NP-hard combinatorial optimization problem [40]. Furthermore, the reliability of network is computationally costly to calculate with the growth of network size, i.e., the calculation of ATR is also NP-hard. These two aspects occurring in the topological optimization problem combine to make it a problem which is extremely difficult to solve, even for problem with small size.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

4.2 Literature Review on the TOP

4.2.1 Calculation of ATR

In order to solve this NP-hard topological optimization problem, it is necessary to calculate the network reliability in the computation. Unfortunately, the calculation of network reliability is NP-hard [110] [88]. In order to calculate the reliability, Satyanarayana and Chang used the Factoring method and solve the reliability in by a recursion algorithm [95]. Page and Perry [82] presented a portable microcomputer implementation for the network reliability using the factoring theorem. Carlier and Lucet proposed a decomposition algorithm for the K-terminal reliability of undirected networks [18]. They showed that their decomposition method can solve much larger networks than the factoring method. And in [16], they showed that the well-known series-parallel reductions can be incorporated in the recursive variance reduction simulation method, leading to a more effective estimator. In [17], Cancela and Urquhart applied the RVR (recursive variance reduction) simulation technique for computing the K-terminal residual connectedness reliability measure in the case of networks where nodes can fail. They proved that an RVR simulation of the residual connectedness reliability has a lower variance than standard Monte Carlo simulation, leading to better estimates.

Recently, Yeh [131] [132] proposed a hybrid Monte Carlo method for estimation of network reliability without needing to know all of the minimal pathsets/cutsets in advance. The analysis in their work indicated that the proposed method is effective when compared with the crude Monte Carlo method. Sekine and Imai proposed a new approach to compute the network reliability function in a mildly exponential time [96] [97], which outperform other exponential-time algorithms based on the recursive formula. In [10], Imai *et al.* reported computational results of the proposed approach by analyzing network reliability against probabilistic link failures. And the results showed that their approach is a fundamental technology for analyzing network reliability in practice.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

4.2.2 Previous Algorithms on solving this problem

Lots of works have been done on this problem. the previous work can be categorized to enumeration-based and heuristic methods. For the former, it can find global optimal solution for the problem, but it's inefficient when applied to large instance of problem. For the latter, it is suitable and feasible to find sub-optimal solutions with a reasonable computation cost.

For the enumeration-based ones, Jan *et al* [55] [56] developed an algorithm using decomposition based on branch-and-bound to find the exact, optimal solution. They divided the problem into several subproblems. Because the network reliability measure is also an NP-hard problem , instead of the exact calculation of reliability, they proposed a upper bound for the reliability. The BnB is able to find the global optimal solution for the TOP, but it is not fit for the large problems due to its exponentially increased computation cost. The maximum network size of fully connected graphs for which a solution has been found using BnB is 12 nodes (60 links). Later on, Koide *et al.* [66] improved their algorithm based on the work of Jan *et al* and extended their algorithm to networks with different edge probability. They also accomplished several improvements to speed up the algorithm in Jan's work and showed the effectiveness of their algorithm.

The heuristic approaches are often adopted due to the NP-hard nature of the TOP. Deeter and Smith [29] used genetic algorithm to solve the TOP. They used a backtracking algorithm [11] to exactly calculate the system reliability. Berna *et al* [31] [32] also adopted genetic algorithms to solve the all-terminal network design problem. They formulate the network design as an integer vector or chromosome. Each gene of the chromosome represents a possible link of the network. In order to deal with the calculation of ATR, instead of calculating the exact reliability, they adopted the estimation method through three steps. First step is the connectivity check for a spanning tree. The next step is the evaluation of a 2-connectivity measure and the last step is to compute Jan's upper bound [56]. A precise estimation

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

of the reliability of the network can be obtained through Monte Carlo simulations.

Recently, neural network approach has proven to be an effective method to solving combinatorial optimization problems. The strength of the neural network approach is that the computational time are very slowly increasing with respect to problem size in solving some kind of NP-hard problems as in [3]. In recent research on the TOP, AboElFotoh and Al-Sumait [3] applied Hopfield-based neural network, which is called optimized neural network (OPTI-nets) in their paper, to solve the topological optimization problem. They formulate each link of the network with a neuron. If the output of a neuron is nonzero, the corresponding link is selected. The simulation results show that the OPTI-net is highly effective in searching optimal or near-optimal solutions in a relative short time. Also the OPTI-net is also very flexible which can not only find good solution for small problems, it also can be applied to a large network with 50 nodes and 1225 edges. The results in their work showed that the neural approach is more effective in designing networks of large sizes compared to other heuristic techniques.

In this chapter, I attempted to use the NCNN to solve the TOP and compare with the OPTI-net and previous methods. I will adopt the upper bound of the network reliability proposed in [3] in the neural network computation. And the correctness of the final solution is verified through the steps used in [31] to ensure that the ATR of the solution is satisfying the reliability constraint. In following Sections, I will introduce the neural network formulation for this problem together with applications and discussions of simulation results.

4.3 Problem Formulation

In this Section, I briefly review the topological optimization problem in network design. A communication network can be modeled by a graph $G = (V, E, P, C)$. Here vertex set V and edge set E are network sites and communication links,

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

respectively. $P = \{p_{ij}\}$ is the set of reliability for all links and $C = \{c_{ij}\}$ is the set of costs for all links. The assumptions for the topological optimization problem are [31]:

1) Each link is bi-directional. No redundant links and cycles exist in the network.

2) Nodes in the network are perfectly reliable and do not fail.

3) The failure of links are mutually statistically independent.

4) No repair is considered.

The optimization problem is:

$$\begin{aligned} \text{Minimize:} \quad & C(x) \\ \text{Subject to:} \quad & R \geq R_0 \end{aligned} \tag{4.1}$$

where R is the all-terminal reliability of the network and R_0 is the network reliability requirement. $C(x)$ is the cost summation of the selected network links, i.e., $C(x) = \sum_{i=1}^{L} c_{ij} x_{ij}$ where L is the total number of links of the network instances and x_{ij} is the binary decision value whether the link is selected in the network defined as below:

$$x_{ij} = \begin{cases} 1, & \text{if link } (i,j) \text{ is selected in the network ;} \\ 0, & \text{otherwise .} \end{cases} \tag{4.2}$$

The all-terminal reliability (ATR) of network G, denoted by R, is the probability that all the vertices in V are connected. Page and Perry [82] used factoring theorem to compute the exact value of network reliability. However, because computing of the ATR is an NP-hard problem [110], [88], estimation of reliability was adopted using various upper bounds [27], [56] and Monte Carlo simulation methods [131]. In this chapter I used the upper bound in eqn. (4.3) to estimate the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

ATR in searching for the optimal solution of the topological optimization problem. The feasibility of the final solution is then verified by computing the exact value of reliability using factoring theorem in [82] for small network instance. And the three steps in [31] are used to deal with the reliability for large network instances, i.e.:

1) perform connectivity check for a spanning tree for the resulting solution.

2) perform the evaluation of a 2-connectivity measure.

3) compute Jan's upper bound [56] for the solution and perform the precise estimation of the reliability of the network using Monte Carlo simulations

The upper bound of ATR used in the network computation is [3]:

$$R(G) \leq \prod_{i=1, i\neq s}^{N} \left[1 - \prod_{j \in A(i)} (1 - p_{ij}) \right], \tag{4.3}$$

where $A(i)$ is defined as the set of vertices connected to vertex i, N is the total number of nodes in the network, p_{ij} is the reliability of link (i, j). Node s is selected as the node with the maximum node degree.

4.4 Applying NCNN to Topological Optimization Problem

4.4.1 Neural Network Formulation

I used a two-dimensional neural network for an N-vertex network instance. The two-dimensional neural network consist of $N \times N$ neurons as shown in Fig. 4.1, where the black square of in the Fig. 4.1 stands for the output of neuron is 1. Each neuron output v_{ij} stands for the selection status of link between vertex i and j same as eqn. (4.2). Because the links in the network instance is bidirectional,

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

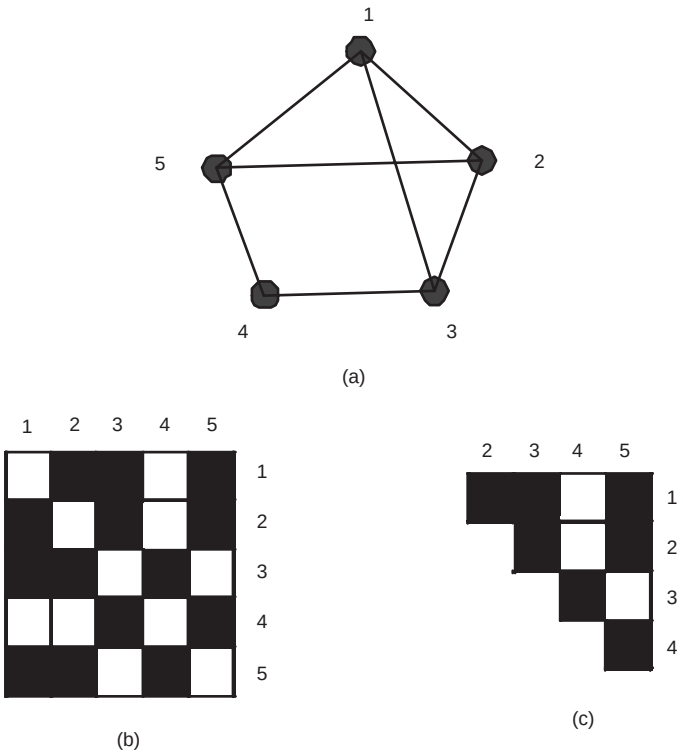


Figure 4.1: A neural network formulation for the optimal solution of benchmark problem 1 (fully connected): (a) the final optimal network with 5 vertices and 7 edges, (b) the $N \times N$ neural network representation for an N-vertex network, and (c) the final neural network representation.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

so the $N \times N$ neurons matrix is symmetrical. It is obvious that only $n(n-1)/2$ elements are needed in the final neuron computation, i.e., the upper triangular of the matrix as shown in Fig. 4.1 (c).

The optimization problem is then formulated as below [31]:

$$\begin{aligned} \text{Minimize:} \quad & \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} v_{ij} \\ \text{Subject to:} \quad & R \geq R_0 \end{aligned} \tag{4.4}$$

4.4.2 Energy Function

I formulated the energy function of the NCNN as follows:

$$E = -W_1 R + W_2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} v_{ij} c_{ij} + W_3 \eta^{\varsigma} \times \left| 1 - \frac{R}{R_0} \right| , \tag{4.5}$$

where $\varsigma = u(1 - \frac{R}{R_0})$ and $u(x)$ is a unit step function, i.e., $u(x) = 0$ for $x < 0$ and $u(x) = 1$ for $x \geq 0$. W_1, W_2, and W_3 are weight factors. The W_1 term encourages the network to increase the network reliability. The W_2 term is the costs of network links to be minimized. The W_3 term is the constraint term which discourages the network from adding more links to increase the network reliability unnecessarily over R_0. η is the penalty factor.

The reliability term in energy function, i.e., R is a estimated value of the reliability. I use an upper bound as in eqn. (4.3) instead of exact calculation in the neural network computation due to the NP-hard complexity for the exact calculation of ATR.

From eqn. (2.13), (2.16) and (4.5), the motion equation of the NCNN is:

$$y_{jk}(t+1) = k y_{jk}(t) + \alpha \left\{ W_1 \frac{\partial R}{\partial v_{ij}} - W_2 c_{ij} - \frac{W_3}{R_0} (-1)^{\varsigma} \eta^{\varsigma} \frac{\partial R}{\partial v_{ij}} \right\}$$

$$- z(t) \left[x_{jk}(t) - I_0 \right] + n(t) , \tag{4.6}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

where from equation (4.3) [3],

$$\frac{\partial R}{\partial v_{ij}} = \prod_{k=1,k\neq s,i,j}^{N} \left[1 - \prod_{l=1,l\neq k}^{N} \psi_{kl}\right] p_{ij} \cdot \left[\prod_{l=1,l\neq i}^{N} \psi_{jl} + \prod_{l=1,l\neq j}^{N} \psi_{il} - 2 \prod_{l=1,l\neq j}^{N} \psi_{lj} \prod_{l=1,l\neq i,j}^{N} \psi_{li}\right] \tag{4.7}$$

and

$$\psi_{ab} \equiv 1 - p_{ab} \times v_{ab} \,. \tag{4.8}$$

4.5 Simulation Results

4.5.1 Selection of Network Coefficients

$$E = \underbrace{-W_1 R}_{\text{A term}} + \underbrace{W_2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} v_{ij} c_{ij}}_{\text{B term}} + \underbrace{W_3 \eta^{\varsigma} \times \left|1 - \frac{R}{R_0}\right|}_{\text{C term}}, \tag{4.9}$$

The selection of the network parameters and coefficients W_1, W_2, W_3 of the NCNN are an important issue related to its efficiency. The network parameters are similar to those used in other problems [118, 116]. The set of network parameters are listed below:

$$\begin{aligned} k = 0.9, \alpha = 0.015, \beta_1 = 0.001, \beta_2 = 0.0001; \\ \epsilon = 1/250; I_0 = 0.65, , z_0 = 0.08, A[n(0)] = 0.001 \,. \end{aligned} \tag{4.10}$$

Although some researchers had adopted the dynamic coefficients instead of fixed ones where the coefficients in the connection weights evolve dynamically along with the system state [122], [87], this dynamic method is too specific which is dependent on problems even for different instances in the same problem. The

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

general way is to choose the fixed coefficients which were determined empirically. The guide line of the selection of coefficients is that the each term in the dynamic equation should be comparable [3]. I rewrite the energy function in eqn. (4.5) and divide to three terms as in eqn. (4.9). I can see that the A-term and C-term are less than 1 and B-term is $O(100 \cdot L)$ when the link costs are chosen from $(1, ..., 100)$, where L is the total number of links. Hence the coefficients of W_1, W_2 and W_3 should be chosen properly to balance the three terms so that they have appreciable weights in the dynamic equation [3]. It should be noted that for different instance the coefficients are also different. One possible set of coefficients for the energy function is:

$$W_1 = 1.5, W_2 = 0.005, W_3 = 1.2, \eta = 10. \tag{4.11}$$

4.5.2 Benchmark Instances and Results

The benchmark problems are adopted from [31] in which the global optimal results for $n < 10$ are verified by using the exact branch-and-bound method [56]. The problems contains 20 different cases. Cases 1 - 17 are problems concerning fully connected networks and cases 18 - 20 are for non-fully connected networks. The comparison of results is listed in Table 4.1, where p is the link reliability probability for each link and R_0 is the objective network reliability.

The simulation of each case was run 50 times using Pentium V 2.5 GHZ desktop and C code. The comparison of results are listed as in the references to which I shall compare our results [56], [31], [3], [66]. Table 4.1 lists the best cost, average cost, and standard deviation for each instance obtained using the NCNN, and compared with the published results in the previous works. From the results, it can be seen that branch-and-bound cannot be applied to fully-connected network problems with node sizes larger than 14 but lead to optimal solutions for small problems. The GAs [31] is somewhat effective but it is strange to see that sometimes it fails to find better solutions. And it also can be seen that the proposed

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 4.1: Comparisons of network cost obtained using the NCNN for benchmark examples from [31] with previous methods, where the results for the NCNN is listed as best cost, average cost, and standard deviation, (Best/Avg/SD) in 50 runs.

Case	N	E	p	R_0	BnB [56]	TOPO [66]	GA [31]	OPTI [3]	NCNN Best/Avg/SD
1	5	10	0.8	0.9	255	255	255	255	255/258.2/0.01
2	5	10	0.9	0.95	201	255	201	201	201/204.1/0.01
3	7	21	0.9	0.9	720	720	720	720	720/723.5/0.02
4	7	21	0.9	0.95	845	845	845	845	845/851.7/0.02
5	7	21	0.95	0.95	630	630	630	630	630/633.3/0.02
6	8	28	0.9	0.9	208	208	208	208	208/209.2/0.02
7	8	28	0.9	0.95	247	247	247	247	247/250.5/0.03
8	8	28	0.95	0.95	179	179	179	179	179/181.2/0.02
9	9	36	0.9	0.9	239	239	239	239	239/244.8/0.07
10	9	36	0.9	0.95	286	286	286	308	286/292.1/0.03
11	9	36	0.95	0.95	209	209	209	209	209/215.3/0.05
12	10	45	0.9	0.9	154	154	156	154	154/165.2/0.06
13	10	45	0.9	0.95	197	197	205	197	197/201.3/0.03
14	10	45	0.95	0.95	136	136	136	136	136/140.2/0.05
15	15	105	0.9	0.95	N/A	262	317	304	296/302.5/0.04
16	20	190	0.95	0.95	N/A	N/A	926	270	202/215.1/0.05
17	25	300	0.95	0.9	N/A	N/A	1606	402	377/283.2/0.07
18	14	21	0.9	0.9	1063	1063	1063	1063	1063/1073.2/0.08
19	16	24	0.9	0.95	1022	1022	1022	1077	1022/1034.6/0.02
20	20	30	0.95	0.9	596	596	596	596	596/599.5/0.08

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

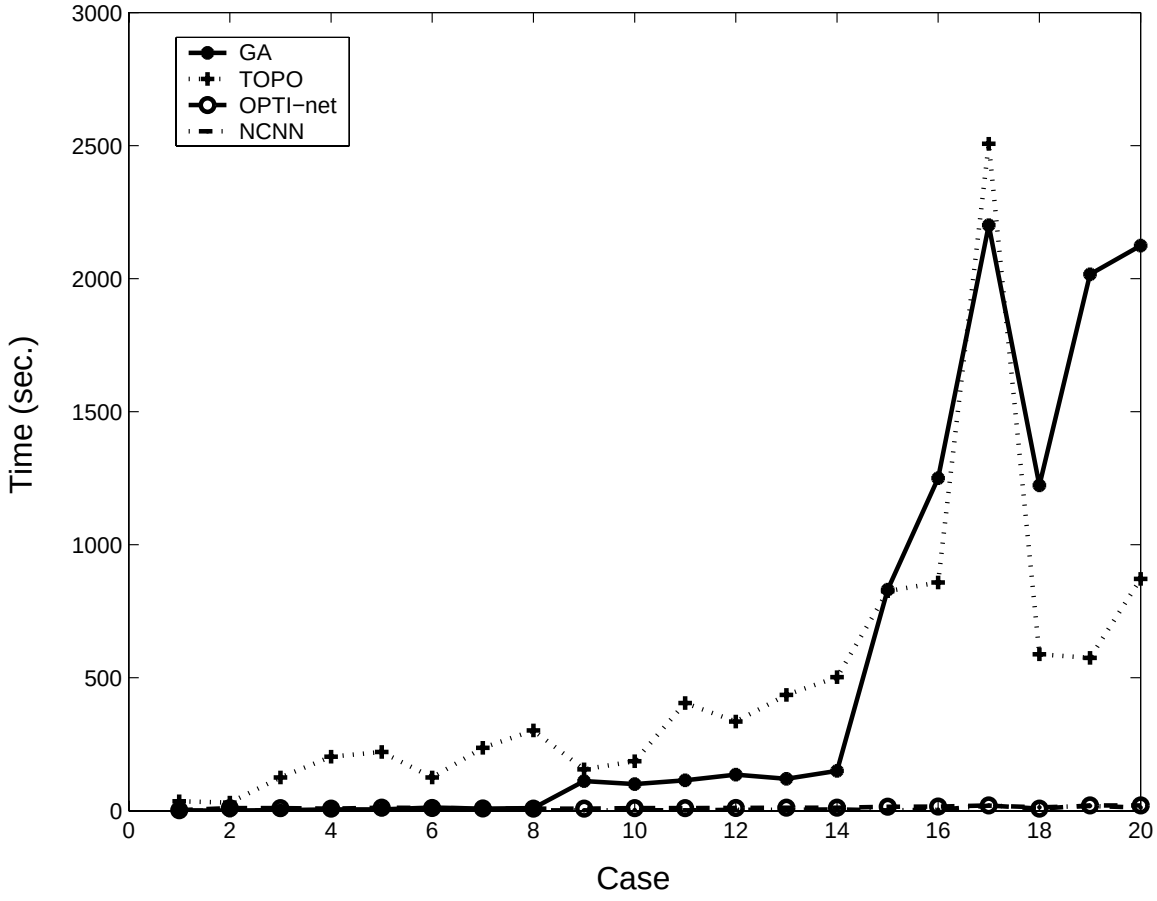


Figure 4.2: Comparison of the computation time required by the proposed method and the previous methods. The x-axis represents the cases from 1 to 20 (the problem size increases from case 1 to case 20), and the y-axis is the computation time required by each case.

NCNN can find equal or better solutions according to the solution quality (cost) compared to the previous methods in all cases.

Paired t-test between the NCNN and the GA method in [31] was performed using the results of 10 runs on case 15. The t value is 13.2. Entering a t table at 9 degrees of freedom (the number of replicate $n - 1$) we find a tabulated t value of 2.26 ($p = 0.05$) going up to a tabulated value of 4.78 ($p = 0.001$). Our calculated t value exceeds these, so the difference between the two compared methods is very highly significant.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 4.2: Comparisons of average computation time (seconds) for benchmark examples with previous methods in 50 runs.

Case	N	E	p	R_0	TOPO [66]	GA [31]	OPTI-net [3]	NCNN
1	5	10	0.8	0.9	1.32	35.52	2.20	0.02
2	5	10	0.9	0.95	1.21	31.23	10.50	1.42
3	7	21	0.9	0.9	2.38	125.44	10.40	1.15
4	7	21	0.9	0.95	5.25	203.18	8.52	0.05
5	7	21	0.95	0.95	5.50	221.35	12.00	0.03
6	8	28	0.9	0.9	12.2	125.24	10.80	1.42
7	8	28	0.9	0.95	8.32	236.57	9.20	1.02
8	8	28	0.95	0.95	10.24	302.26	8.00	1.82
9	9	36	0.9	0.9	111.62	155.80	8.20	1.52
10	9	36	0.9	0.95	100.34	186.38	10.00	2.32
11	9	36	0.95	0.95	114.54	405.28	10.00	2.12
12	10	45	0.9	0.9	136.25	335.78	11.00	3.57
13	10	45	0.9	0.95	120.58	435.32	12.00	4.12
14	10	45	0.95	0.95	150.26	502.46	12.00	5.27
15	15	105	0.9	0.95	831.26	825.58	15.00	1.54
16	20	190	0.95	0.95	1250.23	858.12	16.25	4.25
17	25	300	0.95	0.9	2200.58	2506.62	20.00	19.50
18	14	21	0.9	0.9	1223.12	588.21	8.00	13.42
19	16	24	0.9	0.95	2016.33	575.12	20.12	18.25
20	20	30	0.95	0.9	2124.43	871.25	20.50	12.47

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 4.3: Results of randomly generated network instances

Case	Nodes	Edges	p	R_0	R	Cost	CPU sec
1	30	435	0.9	0.9	0.944±0.02	722±54.9	383.3±0.5
2	40	780	0.9	0.9	0.960±0.01	848±28.6	453±20.7
3	50	1225	0.9	0.9	0.968±0.01	967±8.97	864±168.0
4	60	1770	0.9	0.9	0.986±0.0	1220±12.9	1128±127.0
5	70	2415	0.9	0.9	0.953±0.01	1673±241.0	1270±9.8
6	80	3160	0.9	0.9	0.951±0.02	3916±11.0	3510±219.0
7	90	4005	0.9	0.9	0.960±0.01	3898±401.0	5941±613.0
8	100	4950	0.9	0.9	0.986±0.01	5702±31.8	10390±6.7
9	50	253	0.95	0.9	0.921±0.02	2102±25.3	162±214.0
10	100	501	0.95	0.9	0.954±0.02	6725±21.7	1823±223.0
11	200	1001	0.95	0.9	0.974±0.02	7232±51.2	3625±118.0
12	300	1501	0.95	0.9	0.962±0.03	8771±131.5	6686±268.0

In order to show the scalability of the NCNN, the comparison of the computation time is made and listed in Table 4.2. The values in this table are the average computation time from 50 runs. The problem size increases from the case 1 to case 20. From this table, it can be seen that the time required to obtain a sub-optimal solution is relatively shorter than previous methods. The computational time needed for the NCNN and the previous methods are plotted in Fig. 4.2. From this figure, it can be seen that the NCNN needs relatively shorter time compared with other methods. And the computation time needed grows linearly with the problem size increasing.

Besides the benchmark problem, I evaluate our algorithm on randomly generated big instances with the number of nodes greater than 50 with both fully-connected networks and non-fully connected networks. The number of edges in the non-fully connected networks is around $5N$, where N is the total number of

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 4.4: List of the weighting coefficients for simulated problems

Case	W_1	W_2	W_3
1	1.5	0.002	1.0
2	1.7	0.002	1.0
3	1.6	0.001	1.0
4	1.5	0.001	1.0
5	1.3	0.001	1.0
6	1.6	0.001	1.0
7	1.2	0.001	1.0
8	1.4	0.001	1.0
9	1.4	0.001	1.0
10	1.4	0.001	1.0
11	1.4	0.001	1.0
12	1.2	0.006	1.0
13	1.2	0.06	1.0
14	1.2	0.05	1.0
15	1.3	0.06	1.0
16	1.2	0.04	1.0
17	1.2	0.03	1.0
18	1.0	0.06	1.0
19	1.0	0.09	1.0
20	1.0	0.03	1.0

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

vertices in the network, i.e., $L = 5N$. Our algorithm to randomly generate the non-fully connected network instance is listed as below (It should be noted that the instance generated here should be chosen properly according to the criteria $L = 5N$):

1) Set the number of nodes N of the instance which you want to generate and the edge generation parameter r to be used below. In our simulation, $r = 2/\sqrt{N}$.

2) Generate N two-dimensional coordinates randomly:

$$x_i = Random[0, 1], \quad y_i = Random[0, 1], \quad \text{for} \quad i = 1, ..., N, \tag{4.12}$$

where x_i and y_i are the $x-$ and $y-$coordinates of node i, respectively. And the function $Random[z, w]$ generates a random number uniformly distributed between z and w.

3) Assign an edge for a pair of nodes whose distance is less than r:

$$\begin{aligned} &\text{if} \quad \sqrt{(x_i - x_j)^2 \times (y_i - y_j)^2} < r, \quad \text{then} \quad c_{ij} = 1 \\ &\text{else} \quad c_{ij} = 0\,; \quad \text{for} \quad i = 1, ..., N \text{ and } j = 1, ..., N\,. \end{aligned} \tag{4.13}$$

4) Generated the a random number between $(1, 100)$ as the network link cost. Check the connectivity of the network and output the network structure with connected links and their costs.

The results of the randomly generated instances are shown in Table 4.3. The cases 1 - 8 are fully-connected networks and cases 9 - 12 are non-fully connected ones. The NCNN program was run 30 times for each instance. I used ANOVA (http://www.physics.csbsju.edu/stats/anova.html) to obtain the standard deviations. The error bar of the costs is shown in Fig. 4.3. The results of reliability, cost and running time in Table 4.3 are listed with average $\pm$ standard deviation. It can be seen from the results that the NCNN can found the optimal or sub-optimal solutions in a reasonable amount of time even with large instances. The computation time is relatively small with respect to the size of the network and nearly

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

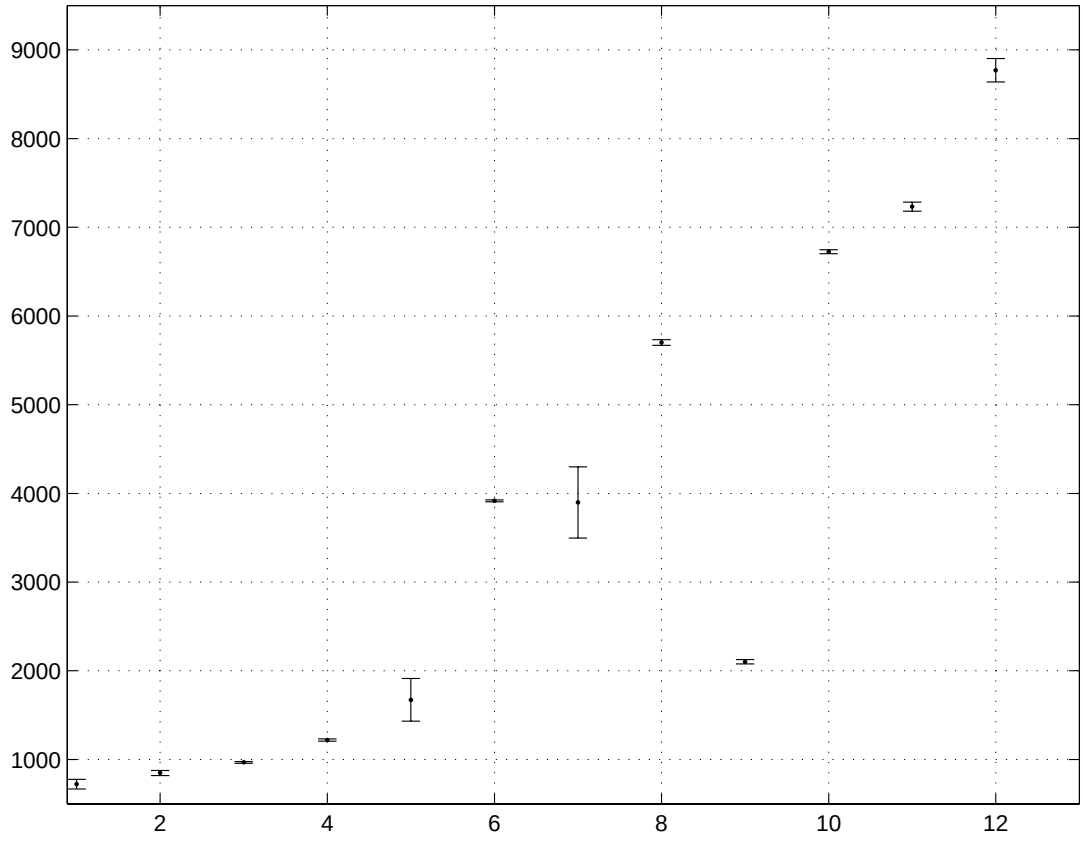


Figure 4.3: Error bars for the cost of the randomly generated network instances. The x-axis stands for each instance and the y-axis is the network cost.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

linear increasing with the problem size. Because case 9 is the non-fully connected instances with relatively smaller number of nodes, the time needed is shorter than fully-connected ones.

4.6 Conclusions

In this chapter, I have used the noisy chaotic neural network to solve the topological optimization problem in backbone networks with all-terminal reliability constraints. The results on 20 benchmark problems showed that our NCNN outperforms other methods, especially in large problems. The results for the randomly generated network with up to 300 vertex showed the computational applicability of the NCNN to solve large-size networks.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 5

Solving the Broadcast Scheduling Problem in Wireless Multihop Networks Using Hybrid Neural Networks

5.1 Introduction

Packet radio networks provide a good option for high-speed wireless data communications, especially over a broad geographic region [75]. Packet radio networks (PRNs) consist of geographically distributed nodes and provides flexible data communication services for nodes through a shared high-speed radio channel by broadcasting. Each node is equipped with a transmitter and a receiver with limited power. Therefore the transmission range is limited and only the nodes within a certain range can transmit messages to or receive messages from each other. If two nodes are far apart, packets need to be relayed by intermediate nodes and traverse several hops before reaching the destination.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The time division multiple access (TDMA) protocol has been adopted in PRNs for nodes to communicate with each other in a single shared radio channel. Since all nodes share one radio channel, conflicts may occur with uncontrolled transmissions, resulting in damaged packets at the destination [133]. These damaged packets increase network delays because they must be re-transmitted. Hence effective broadcast scheduling is necessary to avoid any conflict and to use the limited channel bandwidth efficiently. In a TDMA network, time is divided into frames and each TDMA frame is a collection of time slots. A time slot has a unit time length required for a single packet to be communicated between adjacent nodes. When nodes transmit simultaneously, conflicts will occur if the nodes are in a close range. Therefore, adjacent nodes must be scheduled to transmit in different time slots, while nodes some distance away may be arranged to transmit in the same time slot without causing conflict [112]. The goal of the broadcast scheduling problem (BSP) is to find an optimal TDMA frame structure that fulfills the following two objectives. The first is to schedule transmissions of all nodes in a minimal TDMA frame length without any conflict. The second is to maximize channel utilization or total conflict-free transmissions.

5.2 Previous Methods on Solving the BSP

The BSP has been proven to be an NP-complete combinatorial optimization problem [34], [112] and has been studied in the literatures. Most of the earlier algorithms for the BSP assume that the frame length is fixed and is known *a priori* [34], [39]. Recent algorithms [112] [36] [133] [92] aim at finding both the minimal frame length and the maximum conflict-free transmission, i.e., now there are two objectives in the BSP. Usually, two stages or phases are adopted to tackle the two objectives in a separate fashion. The minimal frame length is achieved in the first stage and conflict-free node transmissions are maximized in the second stage [36] [133] [92].

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Specifically, Ephremides and Truong [34] proposed a distributed greedy algorithm to find a TDMA structure with maximal transmission. They proved that searching for the optimal scheduling of broadcasts in a radio network is NP-complete. Funabiki and Takefuji [39] proposed a parallel algorithm based on an artificial neural network to solve the M-slot problem where the number of time slots is pre-defined. They used hill-climbing to help the system escape from local minima. Wang and Ansari [112] proposed a mean field annealing (MFA) algorithm to find a TDMA cycle with minimum delay time. In order to find the minimal frame length, they first used MFA to find assignments for all nodes with a lower bound of the frame length. For the unassigned nodes left, they then used another heuristic algorithm which adds one time slot at each iteration to the node with the highest degree. The heuristic algorithm was run repeatedly until all the nodes left were assigned. After the number of time slots was found, additional feasible assignments to the nodes were attempted. They also proved NP-completeness of the BSP by transforming the BSP to the maximum independent set problem.

Chakraborty and Hirano [20] used genetic algorithms to solve the BSP. They proposed a GA with a modified crossover operator to handle large networks with complex connectivity. Funabiki and Kitamichi [36] proposed a binary neural network with a gradual expansion scheme, called a gradual neural network (GNN), to find the minimum frame length and the maximum transmissions through a two-phase process. In phase I, they found a valid TDMA cycle to satisfy the two constraints with the minimum number of time slots. The number of time slots in a TDMA cycle was gradually increased at every P iterations from an initial value during the iterative computation of the neural network, until every node can transmit at least once in the cycle without conflicts, where P is a pre-defined parameter. In phase II, additional conflict-free transmissions for the TDMA cycle of phase I were found in order to maximize the total number of transmissions. They demonstrated the performance of their method through the three benchmark instances used in [112] and randomly generated geometric graph instances.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Yeo *et al* [133] proposed a two-phase algorithm based on sequential vertex coloring. They showed that their method can find better solutions compared to the method in [112]. Salcedo-Sanz *et al* [92] proposed a hybrid algorithm which combines a Hopfield neural network for constrain satisfaction and a genetic algorithm for achieving maximal throughput. They partitioned the BSP into two sub-problems, i.e., the problem of maximizing the throughput of the system ($P1$) and the problem to find a feasible frame with one and only one transmission per radio station ($P2$). Accordingly they used a hybrid two-stage algorithm in solving these two sub-problems, i.e., the Hopfield neural network for $P2$, and a combination of genetic algorithms and the Hopfield neural network for $P1$. They compared their results with MFA in [112] in the three benchmark problems, and showed that the hybrid algorithm outperformed MFA.

Recently, there are many hybrid methods proposed to solve the NP-hard COPs, for example [92] [93] [94]. In this chapter, I will propose a hybrid method which combines neural network technique and a sequential vertex coloring and applied it to solve the BSP in two phases with each phase using a different method. A modified sequential vertex coloring algorithm will be applied in the first phase to obtain the minimal number of time slots. And the NCNN model is adopted in the second phases to maximize the node transmissions while meeting the constraints.

5.3 The Broadcast Scheduling Problem

In this section, I will first briefly describe the BSP, as in [39]- [92], and then present a formulation of the BSP based on the description. A packet radio network (PRN) can be represented by a graph $G = (V, E)$, where vertices in $V = \{1, ..., N\}$ are network nodes, N being the total number of nodes in the PRN, and E represents the set of transmission links. Two nodes i and j $(i, j \in V)$ are connected by an undirected edge $e_{ij} \in E$ if and only if they can receive each other's transmission [133]. In such a case, the two nodes i and j are *one-hop* away. If $e_{ij} \notin E$, but

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

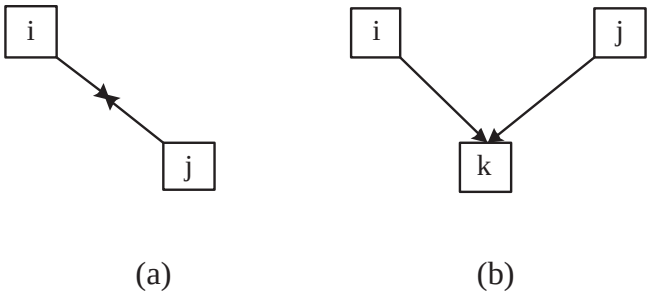


Figure 5.1: Situations in which conflicts occur in a packet radio network. (a) Primary conflict. Node i should not have transmission and reception simultaneously. (b) Secondary conflict. Node k should not receive two or more transmissions simultaneously.

there is an intermediate node k such that $e_{ik} \in E$ and $e_{kj} \in E$, then nodes i and j are *two-hop* away. A *primary conflict* occurs when two nodes which are one-hop away transmit in the same time slot, as shown in Fig. 5.1 (a). A *secondary conflict* occurs if two nodes are two-hop away and transmit in the same time slot, as shown in Fig. 5.1 (b). I summarize the constraints in the BSP in the following two categories:

1) *No-transmission constraint* [20]: Each node in the network should be scheduled to transmit at least once in a TDMA cycle.

2) *No-conflict constraint*: It excludes the primary conflict (a node cannot have transmission and reception simultaneously) and the secondary conflict (a node is not allowed to receive more than one transmission simultaneously).

From the above constraints, It can be seen that two nodes can transmit in the same time slot without conflicts if and only if they are more than two hops away from each other.

The topology of a TDMA network can be represented by an $N \times N$ symmetric

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

binary matrix $C = \{c_{ij}\}(i, j = 1, ..., N)$, called the connectivity matrix:

$$c_{ij} = \begin{cases} 1, & \text{if there is a link between node } i \text{ and node } j, \text{ and } i \neq j\,; \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

An $N \times N$ matrix called the compatibility matrix D $= \{d_{ij}\}$ [36] is defined as follows:

$$d_{ij} = \begin{cases} 1, & \text{if node } i \text{ and node } j \text{ are one-hop away or two-hop away}\,; \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

The final optimal solution is a transmission schedule for all the nodes in the network which consists of M time slots. I will use an $M \times N$ binary matrix $T = (t_{ij})$ to express a transmission schedule [112], where

$$t_{ij} = \begin{cases} 1, & \text{if time slot } i \text{ in a frame is assigned to node } j\,; \\ 0, & \text{otherwise.} \end{cases} \tag{5.3}$$

Channel utilization ρ_j for node j is defined as [112]

$$\rho_j = \frac{\text{the number of time slots assigned to node } j}{\text{TDMA cycle length}} = \frac{\sum_{i=1}^{M} t_{ij}}{M}\,. \tag{5.4}$$

The total channel utilization for the entire network, ρ, is given by [112]:

$$\rho = \frac{1}{N}\sum_{j=1}^{N}\rho_j = \frac{1}{NM}\sum_{j=1}^{N}\sum_{i=1}^{M} t_{ij}\,. \tag{5.5}$$

The goal of the BSP is to find a transmission schedule with the shortest TDMA frame length (i.e., smallest M) which satisfies the above constraints, and at the same time, the total transmissions are maximized. Based on the above description

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

of the BSP, I formulate the BSP as follows:

BSP minimize M and maximize ρ

$$\text{Subject to:} \quad \sum_{i=1}^{M} t_{ij} > 0, \text{ for all } j = 1, 2, ..., N, \tag{5.6}$$

$$\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1, k\neq j}^{N} d_{jk} t_{ij} t_{ik} = 0. \tag{5.7}$$

The no-transmission constraint is formulated in eqn. (5.6), which means each node in the network must transmit at least once in a frame. The no-conflict constraint in eqn. (5.7) indicates that every pair of nodes within one hop or two hops away cannot be scheduled in the same time slot.

A trivial solution satisfying all the above two constraints is an N-slot TDMA cycle where every node transmits in a different time slot. But obviously this solution is not optimal. The BSP is an NP-complete combinatorial optimization problem and to find a global optimum is not easy. In the next section, we will introduce a novel noisy chaotic neural network and then apply this model to solve the BSP in the subsequent sections.

5.4 Hybrid Neural Network Method for the BSP

5.4.1 Phase I: Minimize TDMA frame length using SVC

In this stage, I will aim to minimize the TDMA frame length, which can be reduced to a variation of the classical vertex coloring problem (VCP) [133]. If regarding the colors in the VCP as the time slots in the BSP and vertices in the VCP as nodes in the BSP, the frame length minimization problem becomes the same as the VCP if we require that different slots (colors) must be assigned to one-hop-away and two-hop-away nodes [133]. In this chapter, the phrases of color and time slot are interchangeable, so are node and vertex. The sequential coloring algorithm with

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

vertex ordering is the simplest and practical heuristic algorithm to solve the VCP [25]. In this chapter, I will use the vertex coloring algorithm with a new vertex ordering criteria after the initialization, and we call our algorithm sequential vertex coloring (SVC). The proposed SVC is implemented as follows. A lower bound L_m of the frame length is computed and an initialization procedure pre-assigns a group of nodes to L_m time slots. The vertices are ordered using a traverse algorithm to be described below and the SVC assigns the first available time slot (color) to the first node (vertex). The number of time slots (colors) is increased by 1 when no colors are available for the node. The SVC stops when all nodes are assigned with a time slot.

It is more efficient to search for the frame length starting from a lower bound L_m compared to the value 1. The lower bound L_m can be computed using the following equation [112]:

$$L_m = \max_{i \in V} deg_i + 1 \,. \tag{5.8}$$

where deg_i means the degree of node i in the PRN and the degree of a node is defined as the number of edges connected to this node:

$$deg_i = \sum_{k=1, k \neq i}^{N} c_{ik} \,. \tag{5.9}$$

Note that L_m obtained from eqn. (5.8) is not a tight bound. A tighter bound [92] can be easily obtained using graph theory [63]. Firstly the original graph G = (V, E) is transformed into $G' = (V, E')$, where E in G stands for one-hop-away edges, and E' in G' stands for one-hop-away and two-hop-away edges. The tight bound $L_m^{'}$ is:

$$L_m^{'} = \omega(G') \,. \tag{5.10}$$

where $\omega(G')$ is the maximal cardinality of a clique in G' [92]. In this thesis, I will use the tight bound $L_m^{'}$ as in eqn. (5.10)

The initialization of the SVC is implemented as follows. Firstly I label the group of nodes to be initialized as $G = \{n_i, (i = 1, 2, ...L_m)\}$. The group G consists

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

of the node with the maximal degree and the nodes which are one hop away from this node. Suppose node n_1 is the node with the maximal degree. The other nodes $\{n_i, (i = 2, 3, ..., L_m)\}$ represent nodes one hop away from node n_1. The initial assignments are done as follows: we assign time-slot i to node $n_i, (i = 1, 2, ..., L_m)$. The initialization is based on the fact that all one-hop-away nodes must be assigned with different time slots. The initialization process can reduce the search time, because after the this process, only $N - L_m$ nodes are left to be scheduled.

As stated in [133], different vertex ordering criteria have different effects in searching. In this chapter, in order to take advantage of the initial schedule process, we use the distance criteria, i.e., the topological distance from the pre-assigned nodes to the other unassigned nodes in the graph. It is obtained with a traverse algorithm as follows.

Algorithm traverse (stack S)

1) Set S=$null$. Make a copy of G and name it as Q.

2) Obtain a node x from the queue Q and delete node x from Q. If Q is $null$, then stop and return the node sequence stored in stack S. Else find the one-hop-away nodes of node x denoted as $N_{adj} = \{n_k, (k = 1, 2, ..., p)\}$, where p is the number of one-hop-away nodes.

3) For every node in N_{adj}, if node n_k is not in the queue Q, store the node n_k in Q. If n_k is not in G or queue S, store node n_k in queue S.

4) goto step 2.

Our SVC in the stage I can be described as follows.

Algorithm Sequential Vertex Coloring

1) Find the lower bound using eqn. (5.10). Set the current number of time slots as the lower bound, i.e., $M = L_m$.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

2) Assign time-slot i to node $n_i, (i = 1, 2, ..., L_m)$. Let the counter $p = M + 1$.

3) Order the uncolored vertices in the graph by the distance criterion.

4) Find the number of non-conflicting colors c for the vertex i.

   (a) If $c > 1$, assign the first available color to vertex i.

   (b) If $c = 1$, assign the color to vertex i.

   (c) If $c < 1$, goto Step 6.

5) If p=N, stop; else p=p+1 and goto step 4.

6) $M = M + 1$ and goto step 4.

5.4.2 Phase II: Maximize the channel utilization

The energy function in the second phase can be formulated as the same in [36]:

$$E = \frac{W_1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{\substack{k=1 \\ k \neq i}}^{N} d_{ik} x_{ij} x_{kj} + \frac{W_2}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} (1 - x_{ij})^2 \,. \tag{5.11}$$

where x_{ij} stands for neuron ij. We use $N \times M$ neurons in this phase, i.e., $X = \{x_{ij}\}, (i = 1, ..., N; j = 1, ..., M)$. W_1 and W_2 are weighting coefficients. W_1 represents the relative strength of the constraint term that any pair of nodes which is one hop away or two hops away must not transmit simultaneously during each TDMA cycle. W_2 represents the strength of the optimization term which maximizes the total number of firing neurons or the total node transmissions.

From eqn. eqn. (2.13), (2.16), and eqn. (5.11), we obtain the dynamics of the NCNN for neuron ij as follows:

$$y_{ij}(t+1) = k y_{ij}(t) + \alpha[-W_1(\sum_{\substack{k=1 \\ k \neq i}}^{N} d_{ik} x_{kj}) + W_2 (1 - x_{ij})] - z(t)\,[x_{ij}(t) - I_0] + n(t) \,. \tag{5.12}$$

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

For an N-node BSP problem with M time slots, a noisy chaotic neural network with $N \times M$ neurons are adopted. We convert the continuous output x_{ij} of neuron ij to discrete neuron output t_{ij} as follows.

$$t_{ij} = \begin{cases} 1 & x_{ij} > \sum_{k=1}^{N} \sum_{l=1}^{M} x_{kl}(t)/(N \times M), \\ 0 & \text{otherwise}, \end{cases}$$

The $N \times M$ neurons are updated cyclically and asynchronously in the following sense. All neurons are cyclically updated in a fixed order. After all neurons are updated once, we consider that one iteration is finished. And the new state information is available for the other neurons in the next generation.

The computational complexity of our hybrid method is examined as follows. In the first stage, the complexity for ordering vertex using the traverse algorithm is $O(N+L)$, where N is the number of node and L is the number of edges in the graph. The procedure to find non-conflicting colors in step (4) in the SVC algorithm has $O(N)$ complexity, which needs to run NM times. Thus the complexity in stage I is $O(N^2M)$, where N is the number of node and M is the number of time slot. In stage II, it is difficult to determine the exact number of iterations required by the NCNN for different problem instances with various problem sizes. We investigate the complexity in one iteration step instead: the worst time complexity in one iteration step for the NCNN is $O(NM)$.

5.5 Simulation Results

5.5.1 Parameter Selection

An issue related to efficiency of the NCNN in solving combinatorial optimization problems is to select appropriate parameters including system parameters in the neural network and weighting coefficients in the energy function. Although there are quite a few parameters to be selected in the NCNN, these parameters are

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

similar to those used in other optimization problems [118, 76, 113, 116]:

$$k = 0.9, \alpha = 0.015, \beta_1 = 0.001, \beta_2 = 0.0001,$$
$$\varepsilon = 0.004, I_0 = 0.65, z_0 = 0.08, A[n(0)] = 0.002. \quad (5.13)$$

The weighting coefficients are determined based on the rule that each term (W_1 and W_2) in energy function eqn. (5.11) should be comparable in magnitude, so that none of them dominates. Hence we choose $W_1 = 1.0$ and $W_2 = 0.6$ in the benchmark examples from BM 1 to BM 3. Besides, a comparison of different values of W_2 is carried out for Case 4 to Case 6.

5.5.2 Evaluation Indices

We use three evaluation indices to compare with different algorithms. One is the TDMA frame length M. The second index is the channel utilization factor ρ defined in eqn. (5.5). The third is the average time delay η for each node to broadcast packets [36]:

$$\eta = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{M}{\sum_{j=1}^{M} t_{ij}}\right) = \frac{M}{N}\sum_{i=1}^{N}\left(\frac{1}{\sum_{j=1}^{M} t_{ij}}\right). \quad (5.14)$$

Another definition of the average time delay can be found in [112]. To derive the definition of the average time delay, the following assumptions were made [112]:

(1) Packets have a fixed length, and the length of a time slot is equal to the time required to transmit a packet.

(2) The inter-arrival time for each node i is statistically independent from those of the other nodes, and packets arrive according to a *Poisson* process with a rate of λ_i (packets/slot). The total traffic in node i consists of its own traffic and the incoming traffic from the other nodes. Packets are stored in buffers in each node and the buffer size is infinite.

(3) The probability distribution of the service time of node i is deterministic. Let the service rate of node i be μ_i (packets/slot).

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 5.1: Specifications of benchmark instances.

Instance	Nodes N	edges E	Maximum degree	Minimum degree	Lower bound
BM 1	15	29	7	2	8
BM 2	30	70	8	2	10
BM 3	40	66	7	1	8
Case 4	100	486	18	2	19
Case 5	200	1099	21	2	22
Case 6	400	805	8	2	9

(4) Packets can be transmitted only at the beginning of each time slot.

Under the above assumptions, the PRN can be modeled as N $M/D/1$ queues. According to Pollaczek-Khinchin formula [13], the average time delay for each node D_i is given as follows:

$$D_i = \frac{1}{\mu_i} + \frac{1}{\lambda_i} \frac{(\lambda_i/\mu_i)^2}{2(1-\lambda_i/\mu_i)} . \tag{5.15}$$

where $\mu_i = \sum_{m=1}^{M} x_{mi}/M$ (packets/slot) is the service rate for node i. The total time delay is given by [112]:

$$D = \frac{\sum_{i=1}^{N} \lambda_i D_i}{\sum_{i=1}^{N} \lambda_i} . \tag{5.16}$$

In this thesis, we will use both definitions of the average time delay in eqn. (5.14) and eqn. (5.16) in order to compare with other methods.

5.5.3 Benchmark Problems

Because the NCNN is not applicable in solving the BSP (NCNN is only used in the second phase), the comparisons between the VC-NCNN and the NCNN are

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

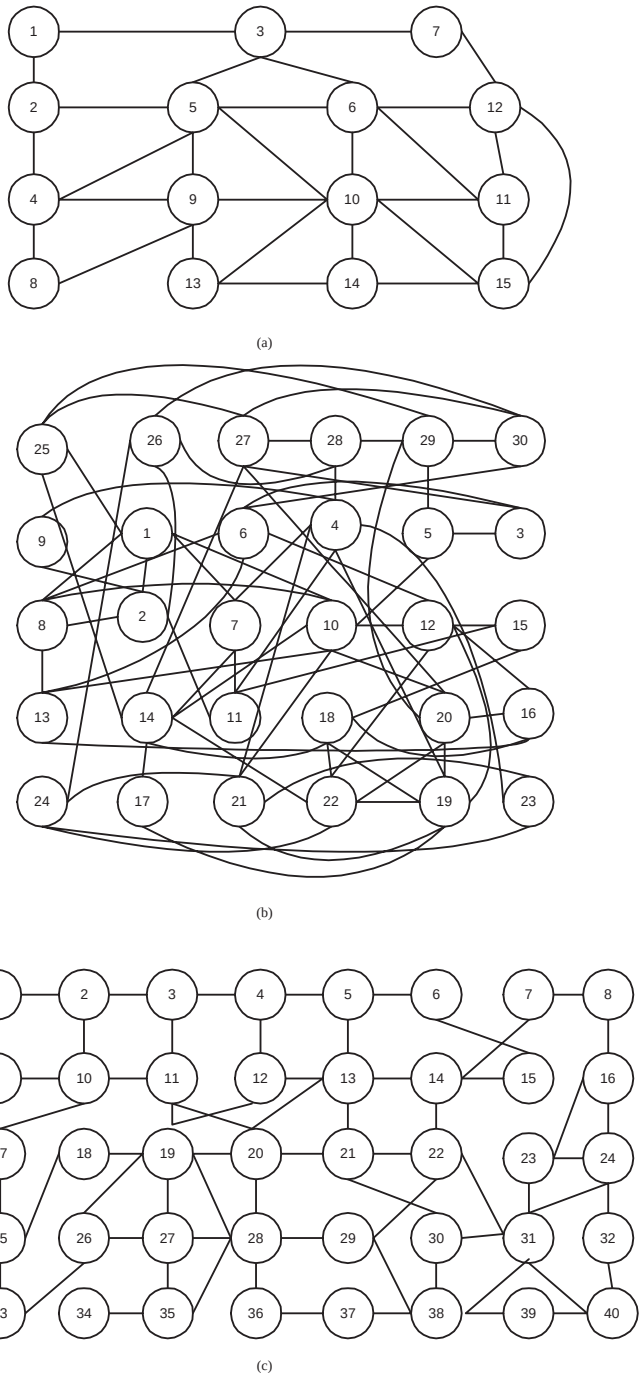


Figure 5.2: The packet radio networks of the three benchmark instances used in [112]. (a) 15-node-29-edge, (b) 30-node-70-edge, and (c) 40-node-66-edge.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

not included. The results will be compared with other previously methods in each simulation problems. We first evaluate our hybrid method through three benchmark problems (BM 1 to BM 3 in Table 5.1), which were solved by the other methods compared in this chapter. The three examples are 15-node-29-edge, 30-node-70-edge, and 40-node-66-edge, respectively. Besides the three benchmark problems, we apply our hybrid algorithm on three large instances (Case 4 to Case 6 in Table 5.1) with the number of node varying from 100 to 400. We run each instance 50 times with different, randomly selected initial states of the NCNN, but with the same set of system parameters as in eqn. (5.13).

The results of the three benchmark problems are summarized in Table 5.2 and compared with other previous methods. Each instance (BM #1 to #3) was run 50 times and shows the best and average results for each problem. Fig. 5.5 plots the curves for each methods compared in this chapter for the benchmark problems. From Table 5.2, we can see that our SVC-NCNN method can find shorter frame length than MFA does. In addition, our proposed hybrid method can find the smallest average time delay η among all methods in all three cases.

The best resulting transmission scheduling for the BM 3 is shown in Fig. 5.3. A black square in this figure represents the transmission of node i in time slot j. From this figure we can see that our proposed method can find feasible schedules for all nodes with the minimal frame length (M = 8) in the first stage. Furthermore, the NCNN can find additional transmissions for each nodes, e.g., node 1 can transmit at additional time slots 4, 5, and 6 in one TDMA frame besides the time slot 1, i.e., node 1 can transmit 4 times in one frame cycle.

In order to see clearly the difference between our method and other methods, I have plotted the average time delay for the three cases as shown in Fig. 5.4. Because no transmission schedules are reported in [36], the GNN method cannot be compared and plotted in these figures. From Fig. 5.4 (a), it can be seen that all other methods obtain the same time delay. But our SVC-NCNN can find better transmission schedule which has the time delays less than other compared methods.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

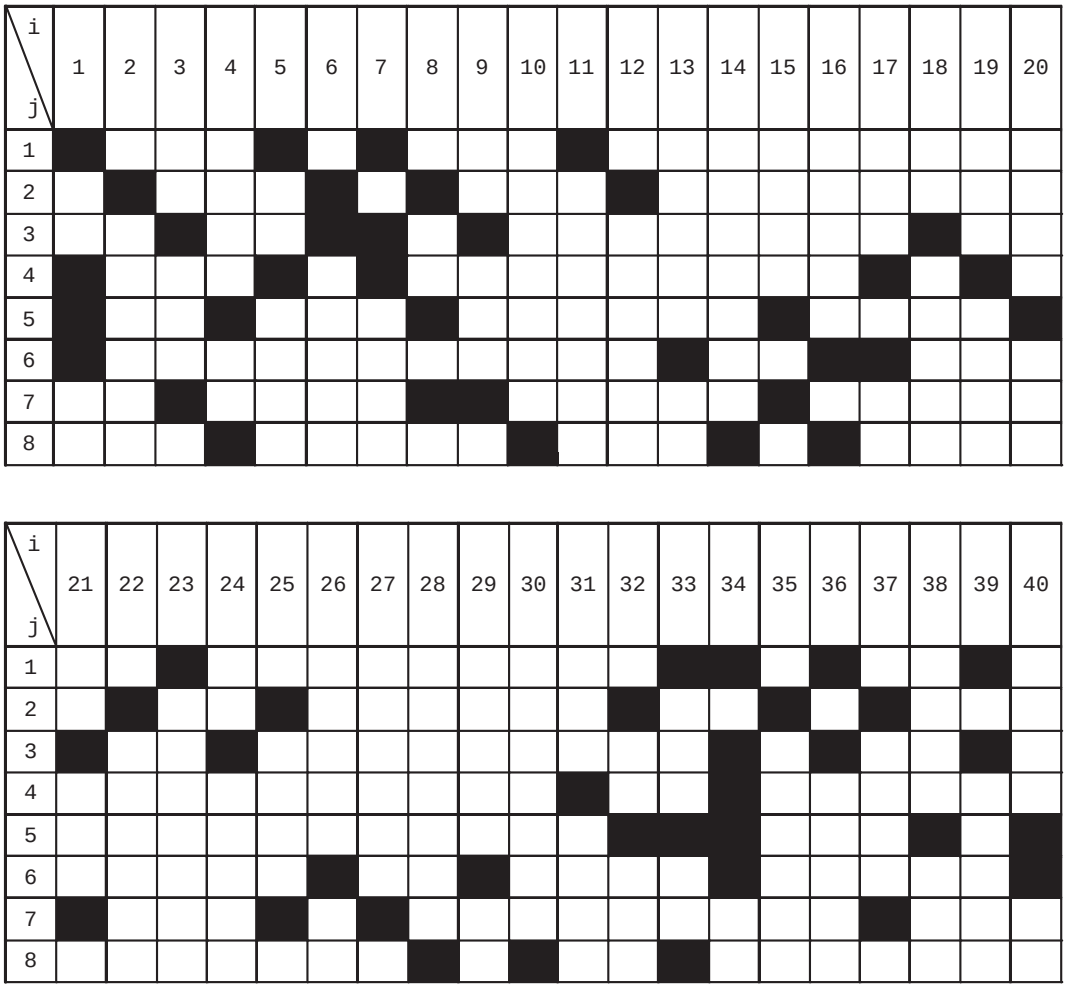
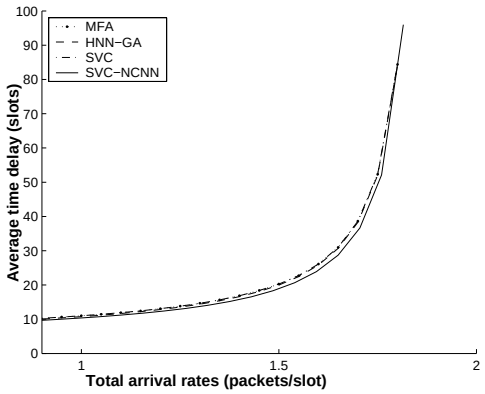
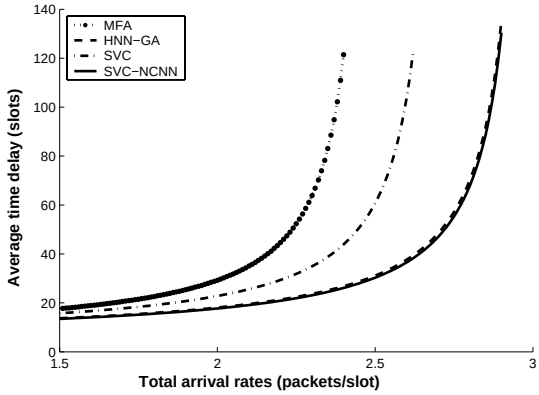


Figure 5.3: The broadcast schedule for the 40-node-66-edge instance. N and M stand for the number of nodes and the number of time slots, respectively. The black square stands for the transmission of node i in slot j.

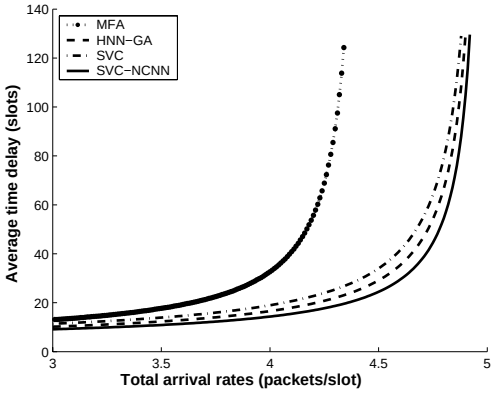
ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(a) BM 1



(b) BM 2



(c) BM 3

Figure 5.4: Comparisons of average time delays for the the three benchmark problems BM 2 and BM 3 (according to Pollaczek-Khinchin formula eqn. (5.16)) with different approaches.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 5.2: Comparisons of best and average results obtained by our SVC-NCNN ($W_1 = 1.0$ and $W_2 = 0.6$) with different algorithms in the three benchmark problems in 50 runs, where η is the average time delay, M is the minimal time slots, and T is the computational time needed (in seconds).

Case		BM 1 Best/Avg	BM 2 Best/Avg	BM 3 Best/Avg
SVC-NCNN	η	6.8/7.0	9.2/9.5	5.8/6.1
	M	8/8.0	10/11.5	8/8.0
	T	3.0/3.5	3.5/4.3	4.0/4.2
HNN-GA	η	7.0/7.0	9.3/10.0	6.3/6.5
	M	8/8.0	10/10.0	8/9.0
	T	4.0/4.7	17/19.0	13.0/14.0
SVC	η	7.2/7.4	10.0/12.0	6.8/7.2
	M	8/8.0	10/11.0	8/10.0
	T	2.5/2.8	15.0/15.4	10.0/12.0
GNN	η	7.1/7.2	9.5/10.0	6.2/6.5
	M	8/8.0	10/10.5	8/8.5
	T	15.0/16.4	18.0/20.0	17.0/19.5
MFA	η	7.2/7.5	10.5/12.5	6.9/8.2
	M	8/9.0	12/13,5	9/10.0
	T	25.0/27.2	32.5/38.5	28.0/29.0

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

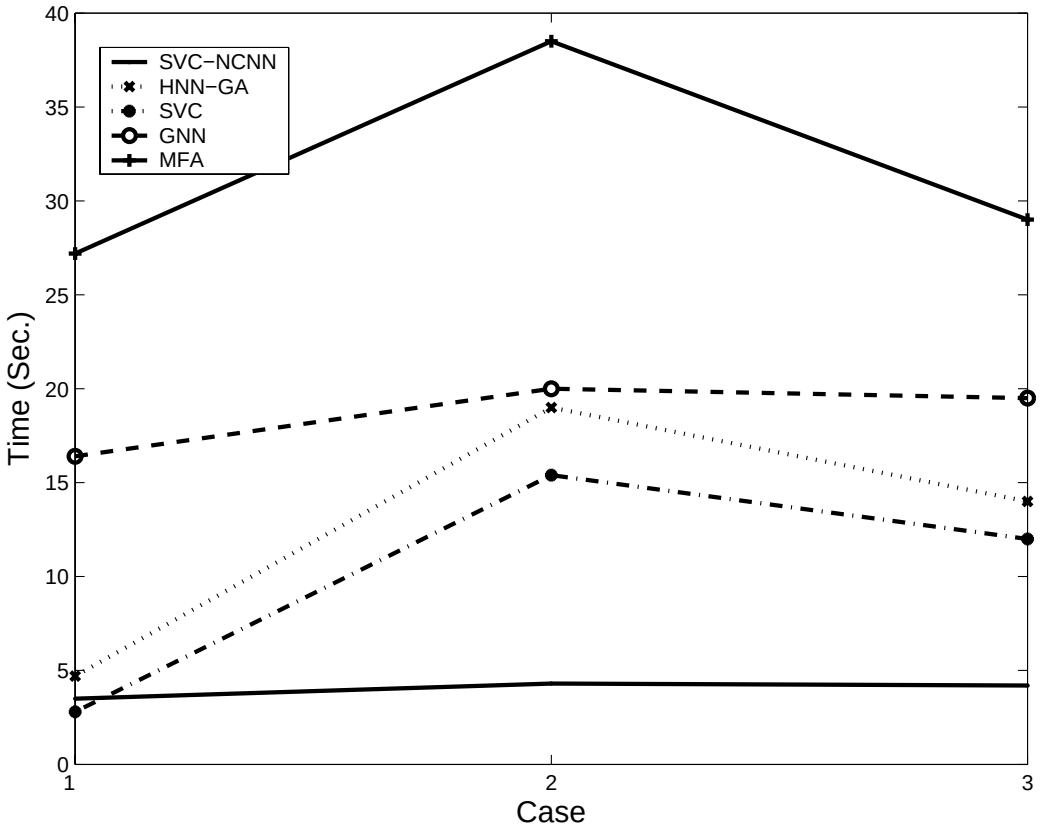


Figure 5.5: Comparison of the computation time required by the proposed method and the previous methods. The x-axis represents the BM 1 to 3, and the y-axis is the computation time required by each case.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

The difference in time delay is greater as shown in Fig. 5.4 (b) and (c) when the problem size becomes large. From Table 5.2 and Fig. 5.4, it can be concluded that SVC-NCNN can find the shortest conflict-free frame schedule while providing the maximum channel utilization among the existing algorithms compared in this chapter.

Paired t-test is performed between the HNN-GA and our proposed methods due the the fact that the curves in Fig. 5.4 are very close. We used the results of 10 runs for BM 1. The t value is 9.7. Entering a t table at 9 degrees of freedom (the number of replicate $n-1$) we find a tabulated t value of 2.26 ($p = 0.05$) going up to a tabulated value of 4.78 ($p = 0.001$). Our calculated t value exceeds these, so the difference between the two compared methods is significant.

In order to show the effect of weighting coefficients in the energy function, I have investigated the effect of different coefficient W_2 in three large instances Case 4 to Case 6. Table 5.3 shows the results of test Case 6 using various values of W_2 while W_1 is fixed. The simulation for each case is run 50 times and list the average values of time slot M, time delay η and runtime (in seconds). Better performance (smaller average time delays) can be obtained with larger W_2, but more iterations are required. When W_2 is greater than 0.9, the algorithm cannot find feasible solutions in the pre-defined 15000 iterations used in our simulations.

5.6 Conclusions

In this chapter, I have proposed a hybrid algorithm which combined sequential vertex coloring and the noisy chaotic neural network to solve the broadcast scheduling problem in wireless multihop networks. I have used two stages for the two objectives of the BSP, i.e., I used sequential vertex coloring to find the minimal TDMA frame length with transmission scheduling in the first stage and the NCNN to maximize channel utilization and minimize time delays in the second stage. I

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 5.3: Simulation results for benchmark example 4 to 6 using our SVC-NCNN on a desktop personal computer with a 2.4GHZ CPU and various choices of W_2 (W_1 is fixed at 1). Here convergence rate is defined as the ratio of the runs in which feasible solutions can be found to all number of runs.

	W_2	M	η	Runtime (sec)	Iteration steps	Converge Rate
Case 4	0.4	20	15.76	19.5	2697.5	100%
	0.6	20	15.56	22.3	4335.2	100%
	0.8	20	15.26	40.5	5894.5	100%
	≥ 0.9	19	-	107.0	15000	0%
Case 5	0.4	22	19.22	87.2	2648.3	100%
	0.6	22	18.28	102.3	4672.5	100%
	0.8	22	17.45	325.4	6538.4	100%
	≥ 0.9	22	-	586.0	15000	0%
Case 6	0.4	9	7.77	178.2	3915.3	100%
	0.6	9	7.70	208.1	4634.9	100%
	0.8	9	7.62	365.4	8321.4	100%
	≥ 0.9	9	-	623.0	15000	0%

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

evaluated our hybrid algorithm in three benchmark examples plus three larger instances. The results showed that our hybrid method finds better solutions than the other algorithms did in the examples.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 6

Solving the Broadcast Scheduling Problem in Wireless Multihop Networks Using the Gradual Noisy Chaotic Neural Network

6.1 Introduction

In this chapter, we will apply the proposed G-NCNN to tackle the BSP. Because the BSP needs to find the minimal number of time slots for the transmission. The gradual expansion scheme can be naturally applied in this situation by letting the number of time slot gradually increasing. In the first phase, the G-NCNN is applied to obtain the minimal TDMA frame length. The number of neurons for the BSP in the first phase is not fixed but varies with the searching process for a minimal number of time slot. The neural network consists of $N \times M$ neurons and starts with a lower bound value of M, where M is the number of time slots needed for the transmission in a TDMA frame. The network will stop searching when a feasible scheduling obtained and goes to the second phase. In the second phase,

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

additional transmissions for nodes are computed by adopting the NCNN model in order to obtain the maximal number of node transmissions. Numerical results show that our G-NCNN outperforms the existing algorithms in both the average delay time and the minimal TDMA cycle length.

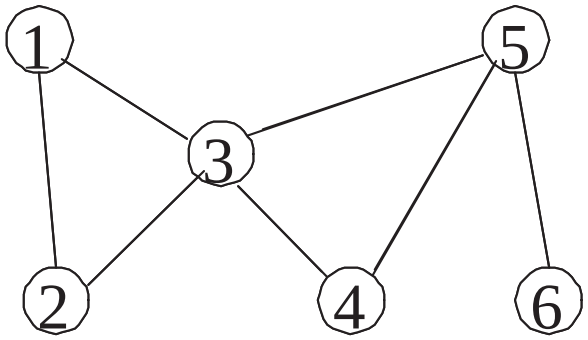
6.2 A Gradual Noisy Chaotic Neural Network for the BSP

As discussed in Chapter 3, the G-NCNN divides the number of neurons into several groups and expands its number of neurons in several stages until the searching stops. In this application, we do not know how many stages needed until we found the solutions. And one of the objectives of the BSP is to find the minimal number of time slots. So we adopt the gradual expansion scheme by adding 1 neuron in each stage. The searching of the G-NCNN is performed in each stage until the feasible solution is found. This application in the BSP can be seen as a special case of the general G-NCNN method by letting the $p = 1$.

6.2.1 Phase I: Minimizing the TDMA Frame Length Using a G-NCNN

For a network with N nodes, we can compute the lower bound of the number of time slot M needed for this network using eqn. (5.10). So for a N-node network, we adopt $M \times N$ neurons where M is the number of time slots and N is the number of nodes. Each neuron ij in the G-NCNN stands for the scheduling of time slot i to node j. The formulation of the BSP in the G-NCNN can be described in Fig. 6.1. In Fig. 6.1 (a) shows an instance of the BSP with 6-nodes and 7-links. From the topology of the network, we can derive the connectivity matrix C in eqn. (5.1) and the compatibility matrix D in eqn. (5.2). The Fig. 6.1 (d) shows the final optimal

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(a) A packet radio network with 6 nodes and 7 links.

C

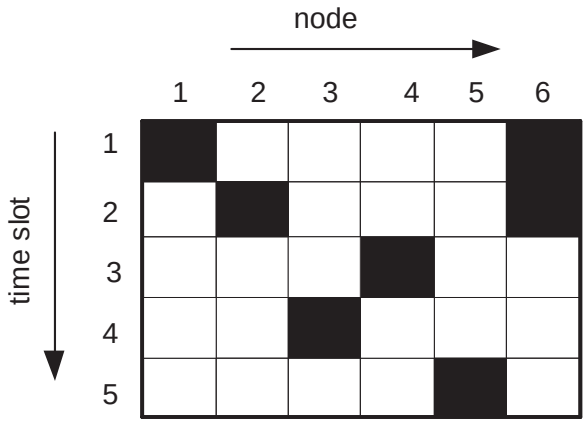
0	1	1	0	0	0
1	0	1	0	0	0
1	1	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	0	0	0	1	0

(b) The connectivity matrix C.

D

0	1	1	1	1	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	0	1
0	0	1	1	1	0

(c) The compatibility matrix D.



(d) The optimum TDMA frame.

Figure 6.1: A neural network formulation for a broadcast scheduling problem.

transmission schedule for this problem, where black square stands for the time slot is assigned to the node, whereas the white square stands for no assignment.

Consider the first objective of the BSP, in order to obtain the minimal number of time slots M, we start to search for solutions with a small M and increase M until a feasible solution is found. The G-NCNN consists of $M \times N$ neurons. M is initially set as its lower bound value L_m as in eqn. (5.10). The GES stops when the G-NCNN finds a feasible assignment and the current number of time slots together with its transmission assignments are the optimal results for phase I of the BSP. In this chapter, different from the GNN in [36], where the neurons are expanded gradually at every P iterations during the iterative computation of the neural network, we implement the GES based on a convergence index $\delta(t)$ of the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

network energy, which we defined as:

$$\delta(t) = \sum_{q=t-4}^{t} |E(q) - E(q-1)| / E(0) . \tag{6.1}$$

where $E(q)$ is the value of energy function at time step q. If index $\delta(t)$ is less than a very small value, e.g., $\delta(t) < 10^{-4}$ in our simulation, the neural network is considered as having fully converged. If the network has converged but no feasible solutions are found using the current number of time slots, the number of time slots is increased by 1, i.e., $M \rightarrow M + 1$, and the G-NCNN re-starts to search for optimal solutions with the updated number of neurons.

The energy function E_1 for phase I is given as follows [36]:

$$E_1 = \frac{W_1}{2} \sum_{j=1}^{N} (\sum_{k=1}^{M} x_{jk} - 1)^2 + \frac{W_2}{2} \sum_{j=1}^{N} \sum_{i=1}^{M} \sum_{k=1, k \neq i}^{N} d_{jk} x_{ij} x_{ki} , \tag{6.2}$$

where W_1 and W_2 are weighting coefficients. The W_1 term represents the constraint that each of the N nodes in the PRN must transmit exactly once during each TDMA cycle. The W_2 term indicates the constraint that any pair of nodes which is one-hop-away or two-hop-away must not transmit simultaneously during each TDMA cycle.

From eqn. (2.13), eqn. (2.16), and eqn. (6.2), we obtain the dynamics of the G-NCNN as follows:

$$y_{jk}(t+1) = k y_{jk}(t) + \alpha \left\{ -W_1 (\sum_{k=1}^{M} x_{jk} - 1) - W_2 \sum_{k=1, k \neq j}^{N} d_{jk} x_{ki} \right\} - z(t) \left[x_{jk}(t) - I_0 \right] + n(t) . \tag{6.3}$$

6.2.2 Phase II: Maximizing Node Transmissions

After phase I, the minimal TDMA frame length M is found and each node is assigned with one and exactly one time slot. In phase II, we aim at maximizing the channel utilization by adding as many conflict-free transmissions as possible

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

to the TDMA frame. Because in phase I one node is assigned with exactly one slot in order to find a minimal frame length, there are many nodes which can use other time slots without violating the no-conflict constraint. Thus, additional transmissions may be found on some nodes but frame length M and the assigned transmissions in phase I are fixed [36]. The second phase here is the same as in section 5.4.2, i.e.,

$$E_2 = \frac{W_3}{2}\sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{k=1,k\neq i}^{N} d_{jk}x_{ij}x_{ki} + \frac{W_4}{2}\sum_{j=1}^{N}\sum_{i=1}^{M}(1-x_{ij})^2\,, \tag{6.4}$$

where W_3 and W_4 are weighting coefficients. W_3 represents the constraint term that any pair of nodes which is one-hop-away or two-hop-away must not transmit simultaneously during each TDMA cycle. W_4 is the optimization term which maximizes the total number of firing neurons.

From eqn. (2.13), eqn. (2.16), and eqn. (6.4), we obtain the dynamics of the NCNN for phase II of the BSP as follows:

$$y_{jk}(t+1) = k y_{jk}(t) + \alpha\left\{-W_3\sum_{k=1,k\neq j}^{N} d_{jk}x_{ki} + W_4(1-x_{ij})\right\} - z(t)\left[x_{jk}(t) - I_0\right] + n(t)\,. \tag{6.5}$$

The NCNN is updated cyclically and asynchronously, which means we update the neurons in two loops and the neuron is selected to be computed in a fixed order. The new state information of a updated neuron is immediately available for the other neurons in the computation. The iteration is terminated once a feasible transmission schedule is obtained, i.e., the transmissions of all nodes are conflict-free.

With regard to the computational complexity of the G-NCNN for the BSP, we noted that previous methods, such as the MFA [112], the HNN-GA [92], and the GNN [36] had no discussions on the issue of time complexity. For the SVC in [133], it is showed that the SVC is a polynomial time algorithm that has $O(N^3)$ computational complexity for the entire search. [38] showed that the worst time

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

complexity in one iteration step is $O(NM^2)$. In phase I, the G-NCNN has the worst time complexity of $O(NM^2)$ for one iteration step. In phase II, the complexity is down to $O(NM)$. Hence, our algorithm has worst time complexity of $O(NM^2)$ in one iteration step. However, it is difficult to determine the exact number of iterations required for different problem instances with various problem size.

6.3 Simulation Results

6.3.1 Parameter Selection

The set of model parameters in eqns. (6.2) - (6.5) had been discussed and listed in eqn. (2.23). The selection of weighting coefficients in the energy function are based on the rule that all terms in the energy function should be comparable in magnitude, so that none of them dominates. Thus we choose the coefficients of the two energy functions as follows:

$$W_1 = 1.0, W_2 = 1.0, W_3 = 1.0, W_4 = 1.0 \tag{6.6}$$

Note that the parameters are chosen experimentally and tuning of these parameters may be necessary when solving different optimization problems or different instances of the same optimization problem, but from our experience, the tuning is only on a small scale. In this chapter, we use the parameters for the benchmark examples as listed in Table 6.1.

6.3.2 Benchmark Problems

Because the NCNN is not applicable in solving the BSP (NCNN is only used in the second phase), the comparisons between the G-NCNN and the NCNN are not included. The results will be compared with other previously methods in each simulation problems. In order to evaluate the performance of our NCNN-based al-

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 6.1: Parameters of the NCNN model for the simulated instances with $W_1 = 1.0, W_2 = 1.0$, and $W_3 = 1.0$.

Case	W_4	k	ϵ	I_0	α	$z(0)$	$n(0)$	β_1	$\beta 2$
1	0.6	0.9	1/250	0.65	0.015	0.08	0.001	0.001	0.0001
2	1.0	0.9	1/250	0.65	0.015	0.08	0.001	0.001	0.0001
3	1.0	0.9	1/250	0.65	0.015	0.08	0.001	0.001	0.0001
1	1.0	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
2	1.0	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
3	1.0	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
4	1.0	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
5	1.0	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
6	0.8	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
7	0.8	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
8	0.8	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
9	0.8	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
10	0.8	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
11	0.8	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
12	0.8	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
13	0.8	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
14	0.8	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
15	0.8	0.9	1/250	0.65	0.015	0.08	0.002	0.001	0.0001
16	0.6	0.9	1/250	0.65	0.015	0.08	0.003	0.001	0.0005
17	0.6	0.9	1/250	0.65	0.015	0.08	0.003	0.001	0.0005
18	0.6	0.9	1/250	0.65	0.015	0.08	0.003	0.001	0.0005
19	0.6	0.9	1/250	0.65	0.015	0.08	0.003	0.001	0.0005
20	0.6	0.9	1/250	0.65	0.015	0.08	0.003	0.001	0.0005

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

gorithm, we first compare it with other methods on the three benchmark problems in [112] shown in Fig. 5.2, which have been solved by all the methods mentioned, i.e., mean field annealing (MFA) [112], the gradual neural network (GNN) [36], the HNN-GA technique [92], and the sequential vertex coloring algorithm (SVC) [133]. We then compare the performance between the G-NCNN and the GNN [36] on a total of 600 randomly generated geometric graph instances. We use the method in [36] to generate the random instances:

1) Set the number of nodes N for the instance to be generated and the edge generation parameter r to be used below. Following [36], we choose $r = 1/\sqrt{N}$ in cases 1 to 5, $r = 2/\sqrt{N}$ in cases 6 to 10, $r = 3/\sqrt{N}$ in cases 11 to 15, and $r = 4/\sqrt{N}$ in cases 16 to 20.

2) Generate N two-dimensional coordinates randomly:

$$x_i = Random[0,1], \quad y_i = Random[0,1], \qquad \text{for} \quad i = 1, ..., N \quad , \tag{6.7}$$

where x_i and y_i are the $x-$ and $y-$coordinates of node i, respectively, and the function $Random[z, w]$ generates a random number uniformly distributed between z and w.

3) Assign an edge for a pair of nodes whose distance is less than r:

$$\begin{aligned} &\text{if} \quad \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < r, \quad \text{then} \quad c_{ij} = 1 \\ &\text{else} \quad c_{ij} = 0\,; \quad \text{for} \quad i = 1, ..., N \text{ and } j = 1, ..., N\,. \end{aligned} \tag{6.8}$$

Each of the 20 geometric graphs are randomly generated 30 times as showed in Table 6.2. As we use the same methods as proposed in [36], our randomly generated graphs are statistically identical to those in [36].

6.3.3 Result Discussions

We use the same evaluation indices as in the previous chapter for the BSP. In this chapter, the three benchmark problems from [112] have been chosen to com-

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

i / j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1					■										
2						■		■							
3	■								■			■			
4							■			■					
5	■							■			■				
6		■										■	■		
7				■			■							■	
8			■					■							■

Figure 6.2: The optimal TDMA frame obtained using the G-NCNN on benchmark instance BM 1.

pare with other algorithms in [112], [36], [133], and [92]. The three examples are instances with 15-node-29-edge (BM 1), 30-node-70-edge (BM 2), and 40-node-66-edge (BM 3), respectively. The optimal TDMA frame obtained using the G-NCNN are displayed in Figs. 6.2 to 6.4, where the i stands for the node and the j represents the time slot. The black square stands for the scheduling of time slot to the specified node.

The dynamics of the G-NCNN energy in phase I of BM 2 are plotted in Fig. 6.5. From Fig. 6.5, we can see clearly that the energy does not decrease smoothly but fluctuates due to the noisy and chaotic nature of the G-NCNN model. The energy gradually settles down to a stable value. The gradual expansion scheme adds the number of time slots by 1 when there are no feasible solutions with the current number of time slots and the G-NCNN restarts the search again. Such re-searching procedure repeats until a feasible assignment is found and the G-NCNN converges.

We labeled our two-phase methods which consists of the G-NCNN in phase

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

i / j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	■														
2					■										
3								■							
4										■	■				
5		■										■			
6									■				■		■
7						■			■					■	
8							■								
9		■	■												
10				■		■									

i / j	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1							■						■		
2				■							■				
3			■					■						■	
4															■
5								■				■			
6		■							■					■	
7															
8					■						■				
9	■					■									
10										■					

Figure 6.3: Same as Fig. 6.2 for BM 2.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

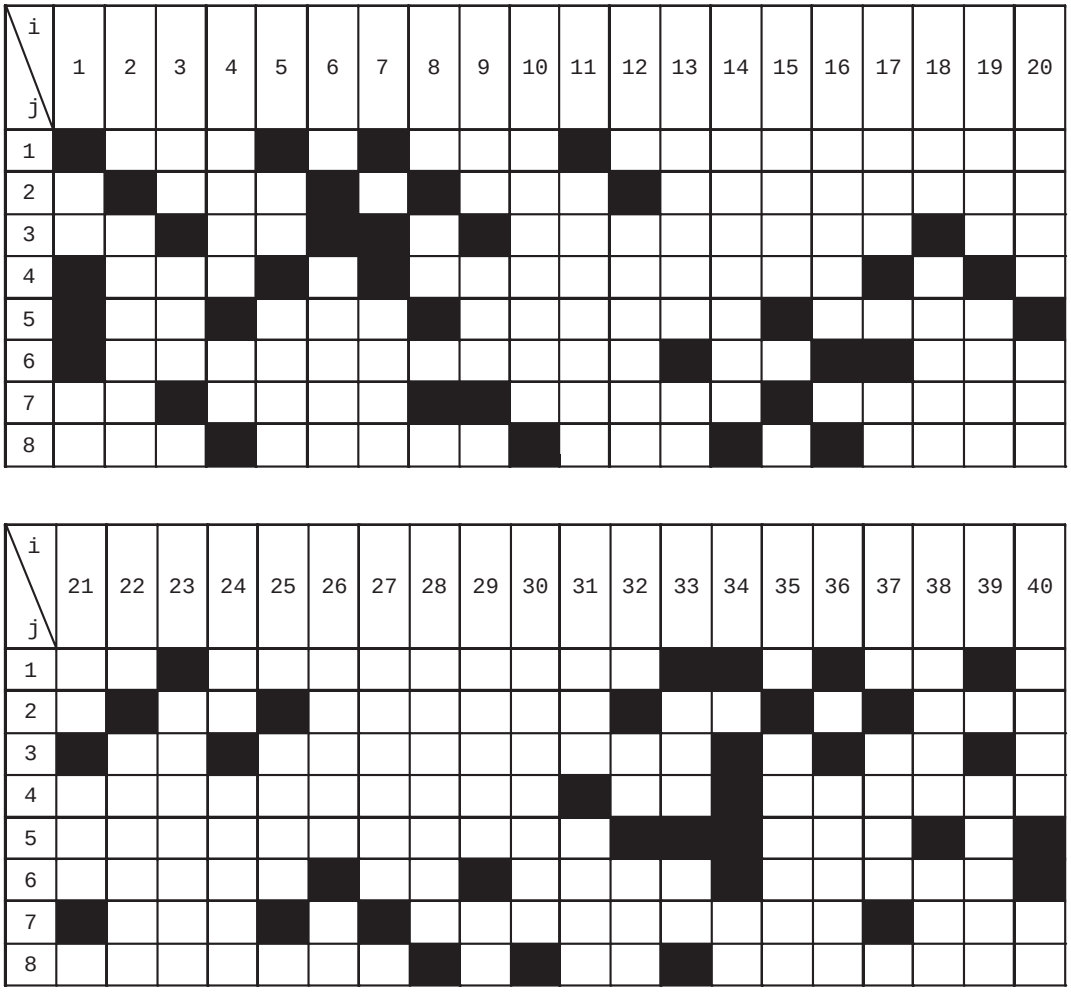


Figure 6.4: Same as Fig. 6.2 for BM 3.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

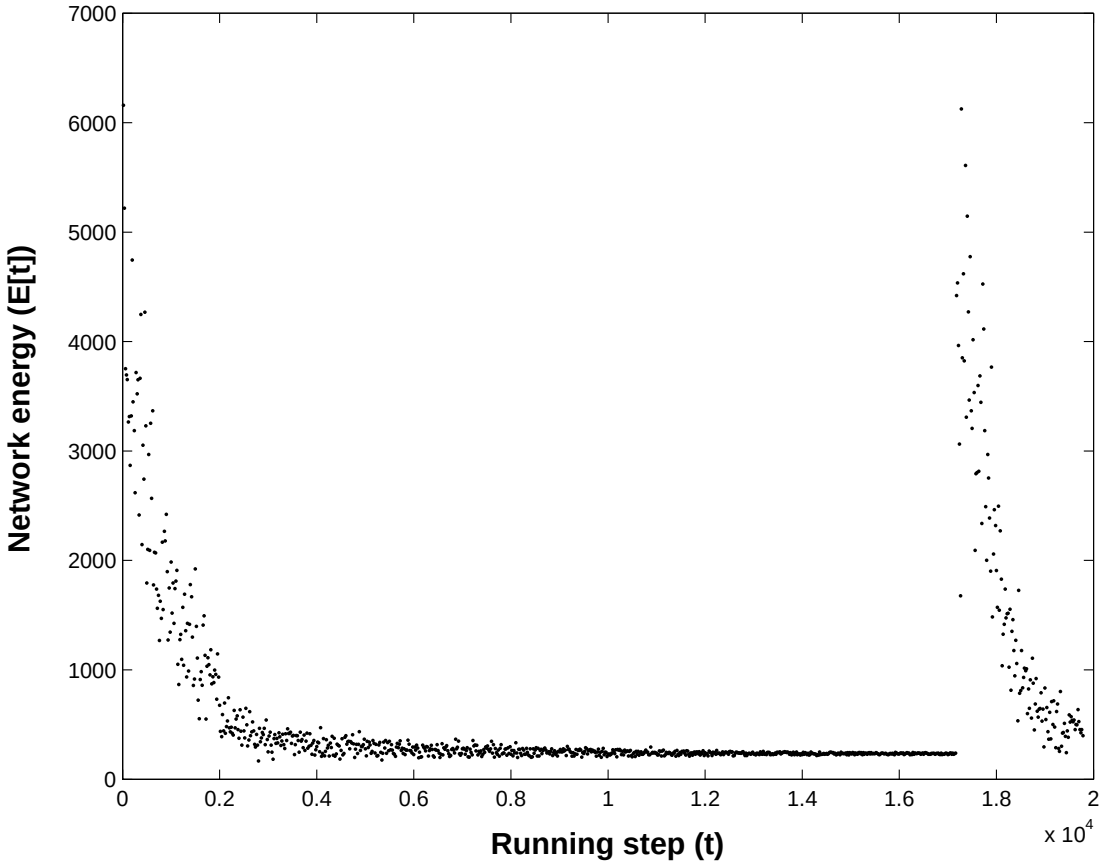


Figure 6.5: Noisy chaotic dynamics of the neural network energy for benchmark BM 2.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

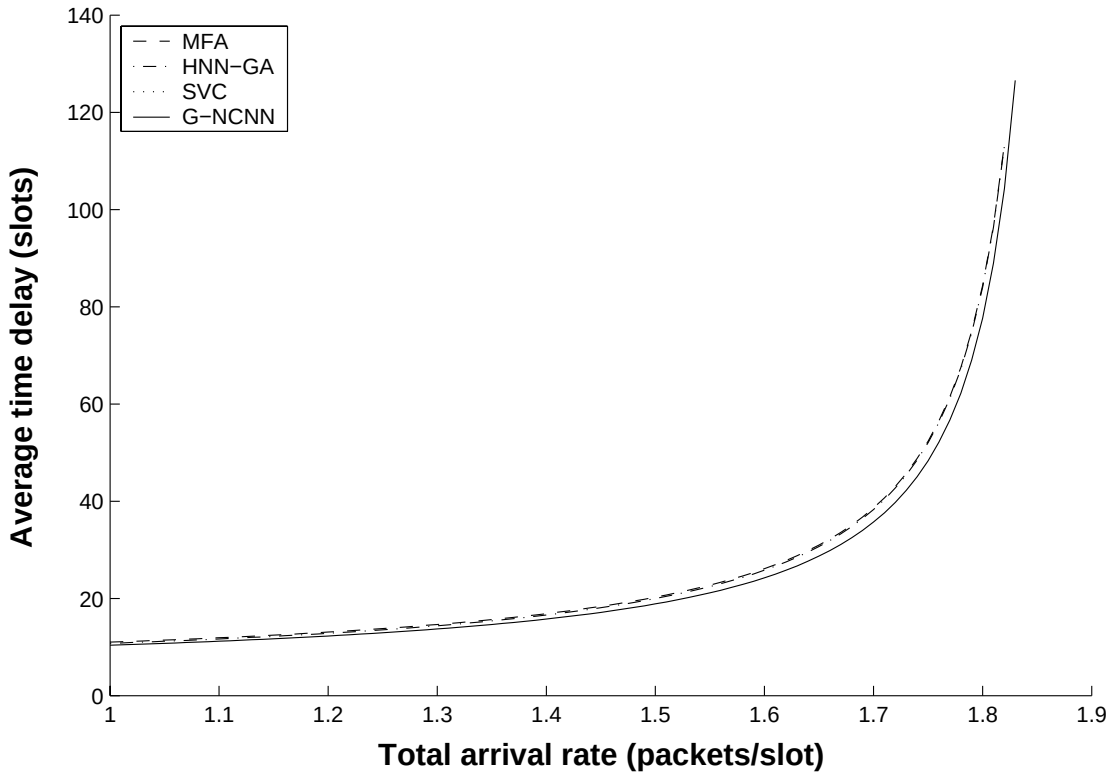


Figure 6.6: A comparison of the average time delay as a function of the total arrival rate for the 15-node-29-edge benchmark (according to the Pollaczek-Khinchin formula given by eqn. (28)) among different approaches.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

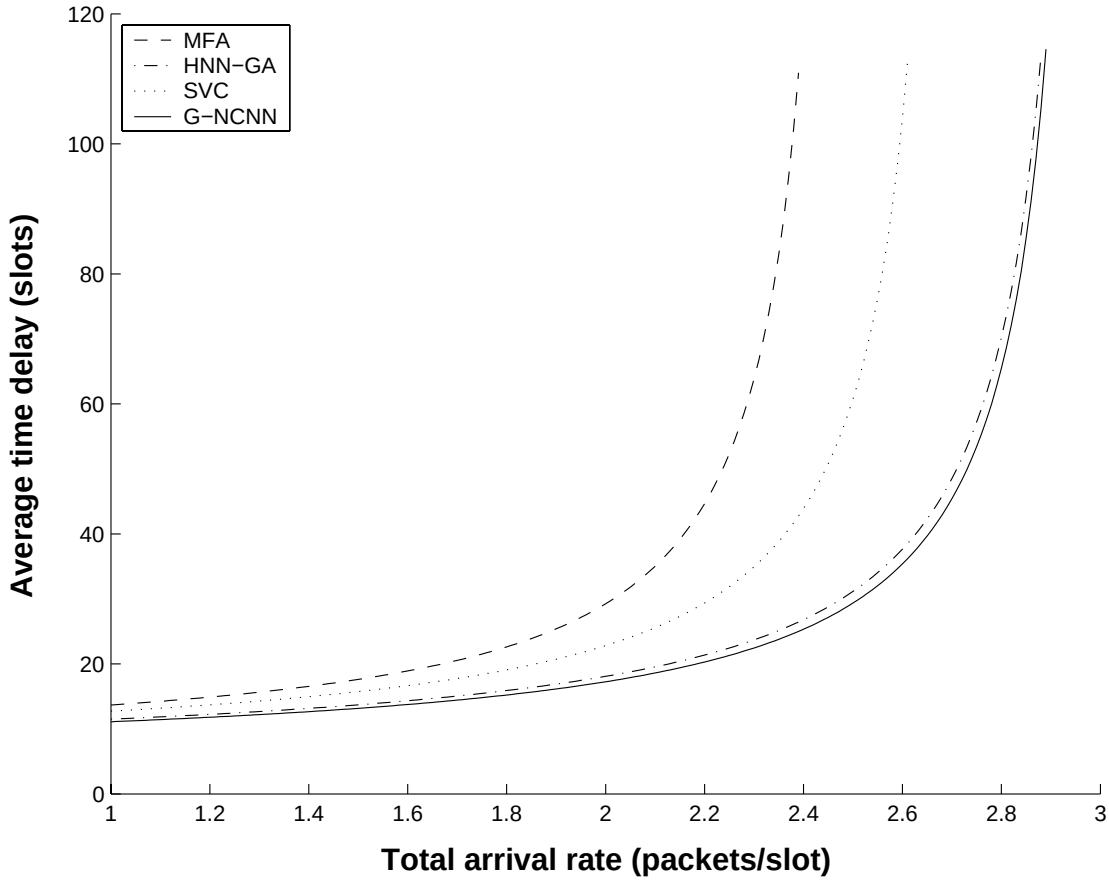


Figure 6.7: Same as Fig. 6.6, for the 30-node-70-edge benchmark.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

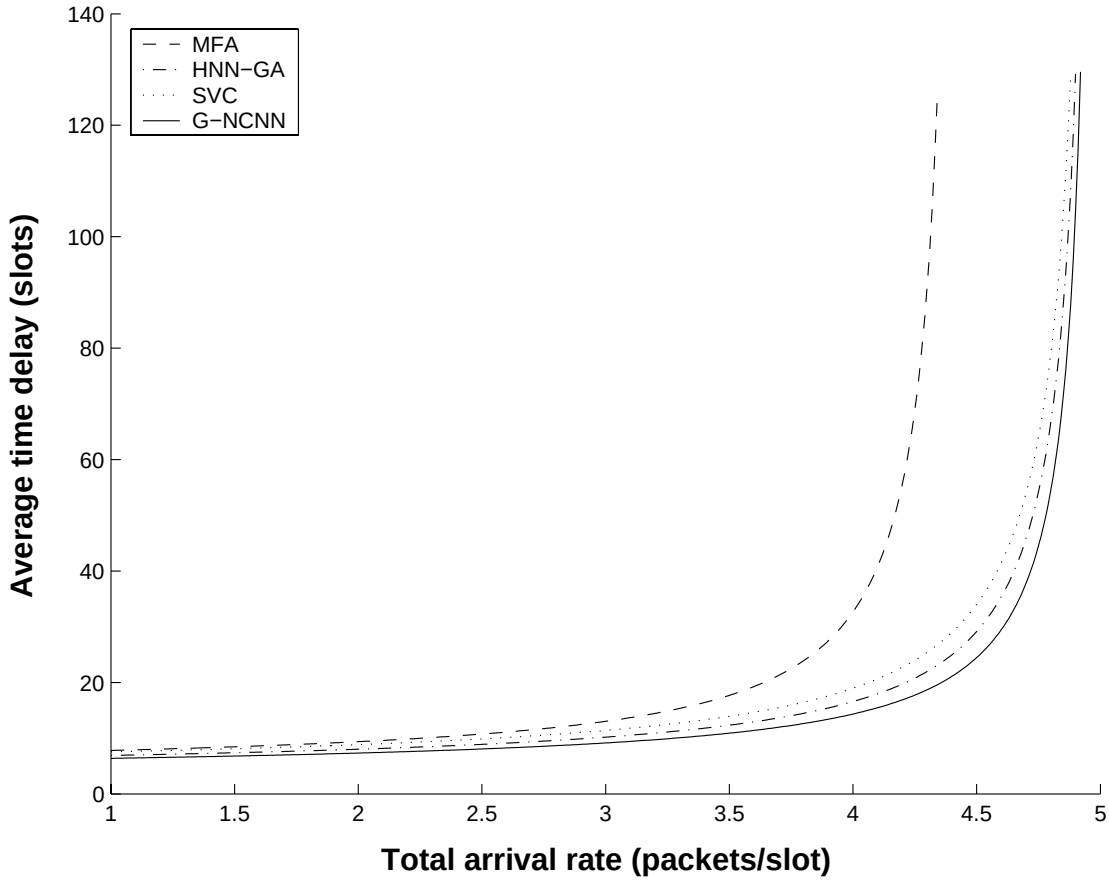


Figure 6.8: Same as Fig. 6.6, for the 40-node-66-edge benchmark.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

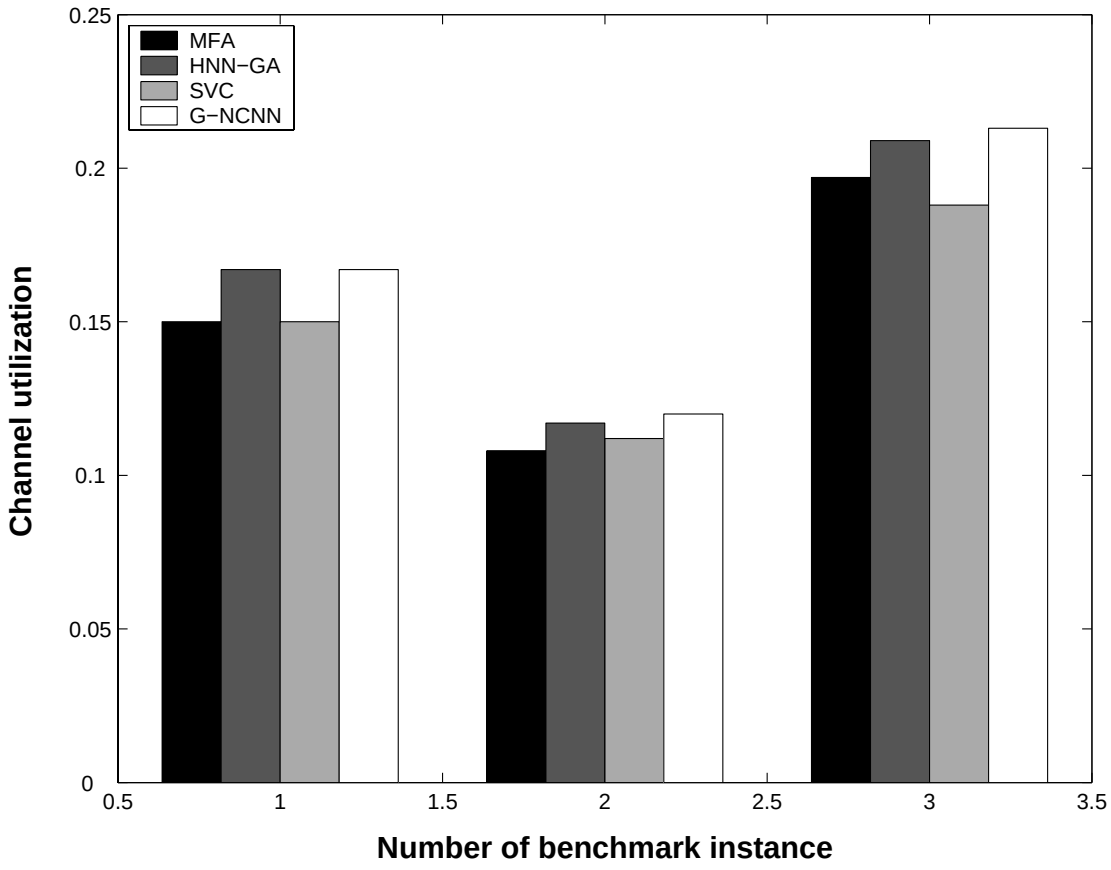


Figure 6.9: Comparisons of channel utilization for the three benchmark problems. 1, 2, and 3 in the horizontal axis stand for benchmark problems BM 1, BM 2, and BM 3 in Table 1, respectively.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 6.2: Specifications of the three benchmark examples given by [112] (BM 1 - 3) and the geometric instances randomly generated as in [36] (cases 1 to 20). The lower bounds for the number of time slots M are displayed as average/maximum/minimum values.

Case	Number of Nodes N	Number of edges E	Maximum degree	Minimum degree	Lower bound on M
BM 1	15	29	7	2	8
BM 2	30	70	8	2	10
BM 3	40	66	7	1	8
case 1	100	150.7	7.3	0.0	8.3/10/7
case 2	300	445.4	8.3	0.0	9.3/11/8
case 3	500	763.3	8.8	0.0	9.8/11/9
case 4	750	1279.8	8.9	2.1	10.0/13/9
case 5	1000	1732.0	9.4	2.0	10.6/14/9
case 6	100	558.0	19.7	1.0	20.7/21/16
case 7	300	1720.5	22.1	2.2	22.7/26/19
case 8	500	2834.5	22.0	2.5	23.0/24/21
case 9	750	4500.2	23.1	2.3	24.5/29/21
case 10	1000	5936.3	25.2	2.0	24.7/28/20
case 11	100	1101.2	33.7	5.9	36.3/43/29
case 12	300	3655.4	40.3	5.4	42.0/48/32
case 13	500	6299.7	42.5	5.7	43.5/50/37
case 14	750	9664.5	43.3	5.9	44.3/51/38
case 15	1000	12993.4	44.2	5.6	45.3/52/40
case 16	100	1755.6	55.4	12.3	57.4/66/46
case 17	300	6168.3	64.3	11.4	66.2/75/59
case 18	500	11201.1	65.3	11.5	68.1/77/62
case 19	750	16621.1	67.0	11.9	68.6/83/62
case 20	1000	22543.2	69.2	13.1	71.6/78/65

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

I and the NCNN in phase II as G-NCNN for short while comparing with other methods. The average time delay for the three benchmark problems are computed based on the final TDMA schedules. Because no schedules have been published on these three benchmark problems in [36], we only compared our G-NCNN with MFA, HNN-GA, and SVC. Figs. 6.6 - 6.8 show that the time delays obtained by the G-NCNN is much smaller compared to those obtained by the MFA algorithm in all three instances. In the 15-node instance, the MFA, SVC and HNN-GA obtained the same time delay, but the time delay obtained by our G-NCNN is less than their results, as we can see from Fig. 6.6. Because BM 1 is a rather small instance with only 15 nodes, the improvement is not much for this case. With increasing problem sizes, improvements made by the NCNN become more evident in BM 2 (Fig. 6.7) and BM 3 (Fig. 6.8). Fig. 6.9 shows that the G-NCNN can find solutions with higher channel utilization compared to MFA, HNN-GA, and SVC.

The computational results for the three benchmark problems are also summarized in Table 6.3. It shows that our G-NCNN can find shorter average time delay as given by eqn. (5.14) in all three benchmark problems compared to the other methods. Compared with the hybrid neural network in the previous chapter, the G-NCNN can find better solution in BM 2, which means the G-NCNN is capable to find better solutions than the hybrid method of SVC-NCNN. Fig. 6.10 plots the curves of the G-NCNN and other compared methods in this chapter. From this figure, we can see that the G-NCNN needs more time compared with the SVC because of its gradual scheme used in finding feasible time slots in the first Phase.

In order to show the difference between the HNN-GA and the NCNN, a paired t-test is performed between the two methods, as shown in Table 6.4. We compared the two methods in 12 cases with node size from 15 to 250, where BM 1 to BM 3 are benchmark examples and case 4 to case 12 are randomly generated instance with edge generation parameter $r = 2/\sqrt{N}$, the mean value of each case is obtained through 50 runs, and the degree of the freedom is 11. The results show that

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 6.3: Comparisons of average delay time η and numbers of time slots M and computation time T obtained by the G-NCNN and other algorithms for the three benchmark problems in 50 runs, where Best/Avg stands for the best value and average value in multiple runs.

Case		BM 1 Best/Avg	BM 2 Best/Avg	BM 3 Best/Avg
G-NCNN	η	6.8/7.0	9.0/9.5	5.7/6.1
	M	8/8.0	10/10.5	8/8.0
	T	6.0/7.2	16.0/18.3	6.0/6.5
HNN-GA	η	7.0/7.0	9.3/9.0	6.3/6.5
	M	8/8.0	10/10.0	8/9.0
	T	4.0/4.7	17/19.0	13.0/14.0
SVC	η	7.2/7.4	10.0/12.0	6.8/7.2
	M	8/8.0	10/11.0	8/10.0
	T	2.5/2.8	15.0/15.4	10.0/12.0
GNN	η	7.1/7.2	9.5/10.0	6.2/6.5
	M	8/8.0	10/10.5	8/8.5
	T	15.0/16.4	18.0/20.0	17.0/19.5
MFA	η	7.2/7.5	10.5/12.5	6.9/8.2
	M	8/9.0	12/13,5	9/10.0
	T	25.0/7.2	32.5/38.5	28.0/29.0

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

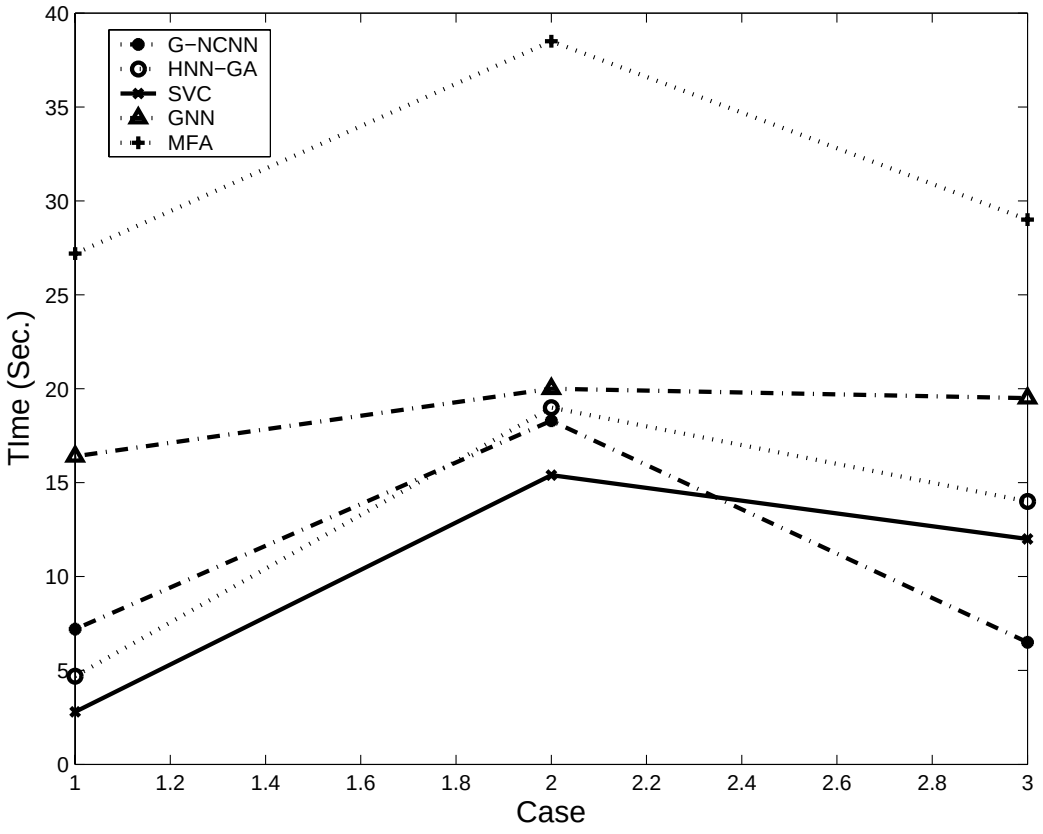


Figure 6.10: Comparison of the computation time required by the proposed method and the previous methods. The x-axis represents the BM 1 to 3, and the y-axis is the computation time required by each case.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 6.4: A paired t-test of average time delay η (second) between the HNN-GA and the G-NCNN.

Case	Node	HNN-GA	G-NCNN
BM 1	15	6.84	6.84
BM 2	30	9.17	9.00
BM 3	40	6.04	5.81
Case 4	60	15.74	13.40
Case 5	80	16.33	14.48
Case 6	100	17.17	15.16
Case 7	120	17.85	16.02
Case 8	150	20.47	16.37
Case 9	180	20.04	16.38
Case 10	200	20.31	17.22
Case 11	230	20.36	16.58
Case 12	250	20.25	17.17
T-Value = 5.22			
P-Value (one-tail) = 0.0001			
P-Value (two-tail) = 0.0003			

the P-value is 0.0001 for one-tail test and 0.0003 for two-tail test. We found that the G-NCNN (mean = 13.7, standard deviation = 4.12) reported having significantly better performance than did the HNN-GA (mean = 15.9, standard deviation = 5.45) did, with T-Value $t(11) = 5.22$, P-Value < 0.05. From Figs. 6.6 - 6.8 and Tables 6.3 - 6.4, it can be concluded that the G-NCNN can always find the shortest conflict-free frame schedule while providing the maximum channel utilization compared to existing algorithms.

In order to show the effects of noise in the computation of the G-NCNN model, we compared the NCNN model with and without noise in three random generated

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 6.5: Results of the TCNN and the NCNN in the 100-node 522-edge benchmark instance using various noise levels in 20 different runs. Here convergence rate is defined as the ratio of the runs in which feasible solutions can be found to all number of runs.

	Noise Level	Minimum Time delay	Average time delay	Standard Deviation	Converge Rate
TCNN	n(0)=0.0	15.38	15.56	0.06	75%
G-NCNN	n(0)=0.001	15.28	15.45	0.16	100%
	n(0)=0.002	15.23	15.57	0.13	100%
	n(0)=0.005	15.16	15.32	0.14	100%
	n(0)=0.009	14.98	15.12	0.10	100%

instance with node size 100, 250 and 750. Tables 6.5 - 6.7 are the results of time delay η for the three instances. The comparisons are performed between the NCNN with different noise amplitude ($n(0) > 0$) and the TCNN with $n(0) = 0$. We ran the simulations of each instance 100 different times using the model parameters and coefficient weights in eqn. (2.22), eqn. (2.23), and eqn. (6.6). "INF" in Table 6.6 and Table 6.7 means that the NCNN algorithms can find feasible solution in the first phase, but failed to converge to a feasible one within pre-defined steps in the second phase, due to the large amplitude of additive noise. "N/A" in Table 6.7 means that the algorithm failed to find a solution even in the first phase of the BSP. From the three tables we can draw the conclusion that both the convergence and the time delay obtained by the NCNN is better than the TCNN in all three instances. The TCNN can find solutions in 100-node and 250-node instance, but when applied to large instances like the 750-node instance, the TCNN failed in all 100 runs. Among different noise levels, it shows that noise level with amplitude $n(0) = 0.002$ has the best performance among all noise levels tested.

Table 6.8 shows the average, maximum, and minimum values of two indices

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 6.6: Same as Table 6.5 for the 250-node 1398-edge benchmark instance. Here convergence rate is defined as the ratio of the runs in which feasible solutions can be found to all number of runs.

	Noise Level	Minimum Time delay	Average time delay	Standard Deviation	Converge Rate
TCNN	n(0)=0.0	17.13	17.15	0.03	90%
G-NCNN	n(0)=0.001	16.73	16.90	0.09	100%
	n(0)=0.002	16.80	16.83	0.05	100%
	n(0)=0.005	16.44	16.52	0.07	100%
	n(0)=0.009	INF	INF	-	0%

η and M of solutions for 600 instances solved by the G-NCNN and the GNN for the 20 randomly generated cases (30 instances for each case). From the results, we can see that the G-NCNN always finds shorter average time delay η than the GNN does in all the simulation cases.

In order to evaluate the stability and scalability of our proposed method, a statistical tool ANOVA, provided as an online services [1] is used to obtain the standard deviations and the error bars for the G-NCNN (Fig. 6.11). However, as the standard deviations for the GNN are not available in their publication, the best results from the GNN is used as a base line to compare with our results (with SD), as plotted in the Fig. 6.11. From this figure, it can be seen that the time delay obtained by proposed G-NCNN is lower than the best results from the GNN in most cases.

The computation time for the proposed G-NCNN is listed in Table 6.9. The algorithm is coded using C++ and simulated on a NEC brand personal computer with 2.4 Ghz CPU and 256M memory. Each benchmark instance was simulated 100 times. The average computational time for each problem together with the

[1] http://www.physics.csbsju.edu/stats/anova.html

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

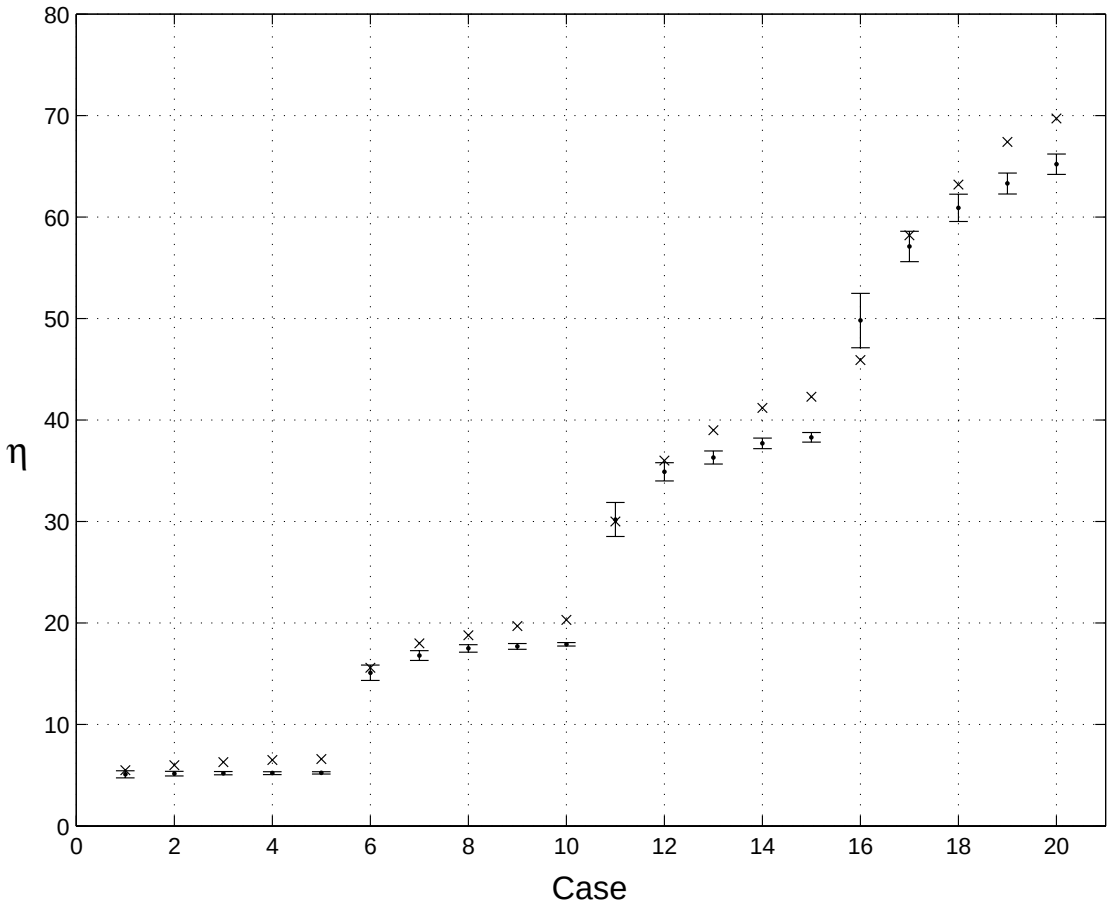


Figure 6.11: Error bars for the average delay time η given by eqn. (26) obtained using the G-NCNN for 20 randomly generated cases of the BSP. The results are average values obtained through 30 different runs. Since the standard deviations for the GNN are not available, only the best (minimum) results for the GNN are plotted here (x-marks).

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 6.7: Same as Table 6.5 for the 750-node 4371-edge benchmark instance. Here convergence rate is defined as the ratio of the runs in which feasible solutions can be found to all number of runs.

	Noise Level	Minimum Time delay	Average time delay	Standard Deviation	Converge Rate
TCNN	n(0)=0.0	N/A	N/A	-	0%
G-NCNN	n(0)=0.001	18.16	18.35	0.11	100%
	n(0)=0.002	18.04	18.20	0.10	70%
	n(0)=0.005	INF	INF	-	0%
	n(0)=0.009	INF	INF	-	0%

standard deviation (SD) were provided in the table. It can be seen that the computation time was increased as the problem size grows. It needed nearly 4 hours for a large problem with 1000 nodes and 6000 edges.

6.4 Conclusions

In this chapter, I have proposed a gradual noisy chaotic neural network model (G-NCNN) for solving the broadcast scheduling problem (BSP) in packet radio networks. A two-phase optimization have been adopted to solve the two objectives of the BSP with two different energy functions. The G-NCNN was adopted to find an optimal transmission schedule with the minimal TDMA frame length in the first phase. In the second phase, additional node transmissions were found using the NCNN based on the results from the first phase.

We evaluated our G-NCNN algorithm in three benchmark examples and 600 randomly generated geometric graph instances. We compared our results with existing methods including mean filed annealing, HNN-GA, the sequential vertex coloring algorithm, and the gradually neural network. The results of the bench-

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 6.8: Comparisons of average delay time η given by eqn. (26) and time slot M obtained by the G-NCNN and the GNN on the randomly generated geometric graph instances. The results are displayed as average $\pm$ standard deviation / maximum / minimum.

Case	G-NCNN		GNN	
	η	M	η	M
1	5.1±0.35/5.8/4.3	8.4/11/6	6.3/7.5/5.5	8.1/11/7
2	5.2±0.224/5.6/4.7	9.3/12/8	6.7/7.5/6.0	9.6/12/8
3	5.2±0.159/5.6/4.7	10.0/12/8	6.8/7.4/6.3	10.0/12/9
4	5.2±0.139/5.5/5.0	10.1/12/9	6.9/7.5/6.5	10.2/13/9
5	5.2±0.112/5.5/5.0	9.8/12/9	7.0/7.5/6.6	10.6/12/9
6	15.1±0.756/16.5/13.8	19.1/23/16	17.4/20.0/15.6	20.2/24/18
7	16.8±0.484/18.2/16.1	22.5/25/21	19.7/20.8/18.0	23.3/26/20
8	17.5±0.374/18.2/17.0	23.1/28/21	20.3/22.7/18.8	24.2/30/22
9	17.7±0.288/18.2/17.2	25.1/28/23	20.9/22.9/19.7	25.4/29/23
10	17.9±0.172/18.3/17.6	25.2/29/24	21.2/22.2/20.3	25.5/28/24
11	30.3±1.67/33.4/26.7	34.3/42/29	34.1/38.2/30.0	37.4/46/32
12	34.9±0.897/37.1/33.3	42.9/48/38	39.1/41.9/36.0	43.2/48/38
13	36.3±0.644/37.6/35.1	43.3/47/40	41.0/44.8/39.0	45.2/52/42
14	37.7±0.512/38.7/36.8	46.0/52/43	42.9/45.4/41.2	47.3/53/45
15	38.3±0.477/39.3/37.5	47.2/52/44	43.7/46.2/42.3	48.1/52/46
16	49.8±2.68/54.5/45.8	56.3/66/46	54.5/60.0/45.9	57.9/65/48
17	57.1±1.50/59.7/54.3	67.6/80/61	63.6/71.5/58.2	68.4/79/61
18	61.0±1.34/63.1/58.4	71.5/81/66	67.4/72.1/63.2	72.2/79/66
19	63.3±1.03/65.4/61.3	73.7/81/70	70.3/75.4/67.4	75.2/83/71
20	65.1±1.01/68.5/63.3	76.8/82/72	72.9/75.0/69.7	78.2/82/73

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 6.9: The computation time for the benchmark instances run on a personal computer with a P4 2.4 GHz CPU (average±standard deviation) seconds

Case	G-NCNN (second)
BM #1	2.0 ± 0.0
BM #2	81.0 ± 0.0
BM #3	6.0 ± 0.0
case #1	9.3±4.7
case #2	159.0±81.1
case #3	514.0±72.0
case #4	1265.0±58.0
case #5	2349.0±42.0
case #6	26.6±10.2
case #7	1255.0±55.0
case #8	3400.6±41.0
case #9	7304.4±112.0
case #10	14100.0±595.0

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

mark problems showed that the G-NCNN always finds the best solutions with minimal average time delays and maximal channel utilization among the existing methods. We showed that our G-NCNN has significant improvements over the HNN-GA through a paired t-test. We also compared the NCNN with the TCNN in three instances and showed that the NCNN has better performance than the TCNN.

I have provided the computational complexity analysis for the G-NCNN, and the computation time for the method are also provided. As I have no benchmark source code for other methods, the computation time comparison with other methods are now not available, and I will make a thorough comparison in the future work in this issue.

The results on this problem in this chapter, together with the results for other optimization problems [118, 76, 113, 116] support the conclusion that the G-NCNN is an effective approach for solving large-scale combinatorial optimization problems.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 7

Solving the Frequency Assignment Problem in Satellite Communications Systems Using the Noisy Chaotic Neural Network with Variable Threshold

7.1 Introduction

Communications satellites make the global village possible and are considered as a billion-dollar technology, with applications ranging from weather forecasting to mobile telecommunications [121]. Due to the economic effect on the average person in everyday life, there is an increasing number of satellites in geostationary orbits. In order to accommodate the crowded satellites in the same orbit, optimal design of satellites are necessary in order to provide high quality transmissions. In satellite communication systems, the major impairments in transmission design include thermal noise, rain attenuation, inter-modulation, and co-channel interfer-

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

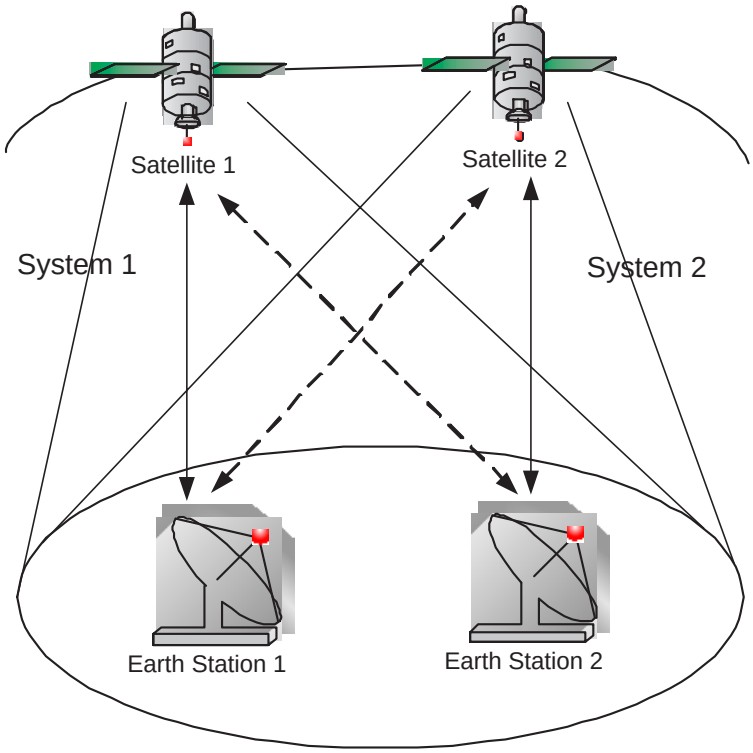


Figure 7.1: Inter-system co-channel interference.

ence, among which, co-channel interference dominates because it seriously affects system design and operation [78]. Hence, the reduction of the co-channel interference has arisen as a major problem in satellite communications with the dramatic increase of geostationary satellites in orbits.

There are two types of co-channel interference [78], i.e., inter-system interference and intra-system interference . The inter-system interference is shown in Fig. 7.1. The sidelobe emission of the earth station antenna gives rise to the inter-system co-channel interference when adjacent satellite systems are operated in the same communication frequency band. Intra-system co-channel interference can be commonly observed in frequency re-used systems through orthogonal polarization and/or multiple spot beams. Radiation pattern of spot beams or polarization discrimination determines the isolation among beams sharing the same frequency band. Co-channel interference is inevitable since perfect isolation cannot be guaranteed [78]. As shown in Fig. 7.2, transponder 1 has beam connection B_I to B_K

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

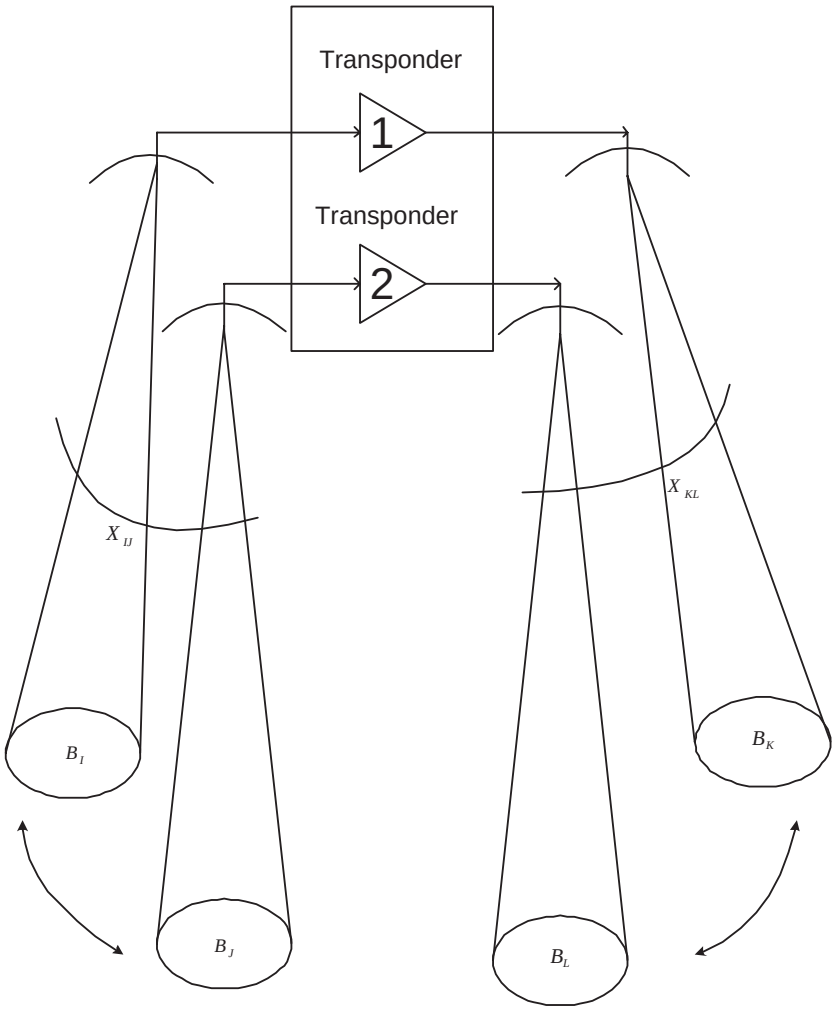


Figure 7.2: Intra-system co-channel interference, where the B_I stands for the beam I. B_j, B_k, and B_L have the same meanings.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

and transponder 2 has beam connection B_j to B_k. Transponder 1 and 2 share the same frequency band. X_{IJ} is the uplink beam isolation between B_I and B_J, and X_{KL} represents the downlink isolation between beams B_k and B_L.

In order to deal with either case of interference, re-arrangements of frequency assignments, which take advantage of carrier interleaving, is thought as an effective way in practical situations. Early efforts have focused on various analytical methods for evaluations of co-channel interference [86] [59] and very few systematic methods have been adopted to optimize frequency assignments to reduce co-channel interference. The later work of Muzuike and Ito [78] revealed the importance of mathematical models for reduction of co-channel interference. They proposed a basic mathematical model to formulate the co-channel interference reduction problem as the "assignment problem". The assignment problem aims to minimize the largest interference among carriers. Fig. 7.3 shows the co-channel interference model for the system in Fig. 7.1. In the inter-system context, the two sets of carriers share the same frequency band. One set of carriers (C_{11} to C_{13}) is in satellite system 1 and the other set of carriers (C_{21} to C_{24}) corresponds to satellite system 2 in Fig. 7.1. In the context of intra-system, system 1 is considered as transponder 1 and system 2 represents transponder 2 in Fig. 7.2.

In the model shown in Fig. 7.3, carrier frequencies for one set of carriers are to be rearranged while keeping the other set fixed, i.e., the frequencies for carriers in system 2 are chosen to be re-arranged while the frequencies for system 1 are fixed. Fig. 7.1 shows the inter-system co-channel interference between two adjacent satellite systems. The communications are assumed to operated between F_a and F_b as shown in Fig. 7.3, where F_a and F_b are frequency band. The co-channel interference can be evaluated by calculating the each pair of carriers using the same frequency, which varies with different pairs. The objective of this assignment problem is to find the optimal assignment of frequencies in system 2 in order to reduce the co-channel interference. The largest interference is considered as a limiting factor, and the optimal assignment is the one which can minimize the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

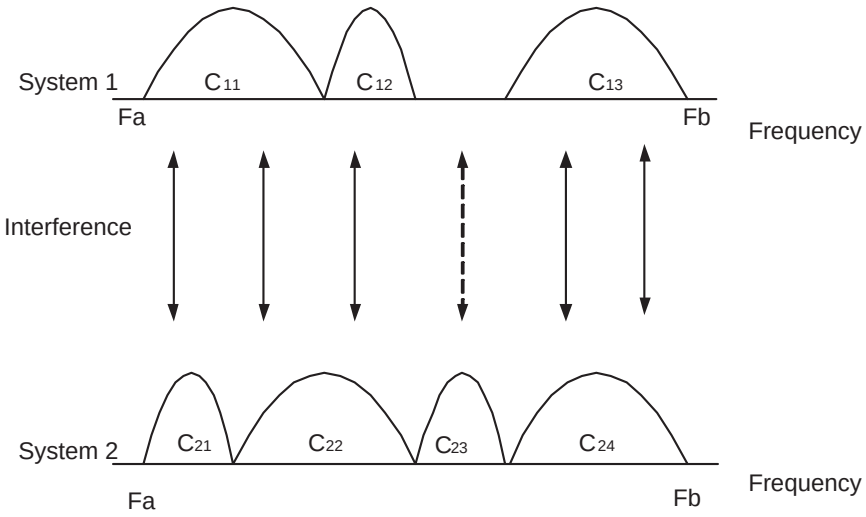


Figure 7.3: Co-channel interference model for the system in Fig. 7.1, where the C_{xy} stands for the carrier y in system x, e.g., C_{23} stands for the carrier 3 in system 2.

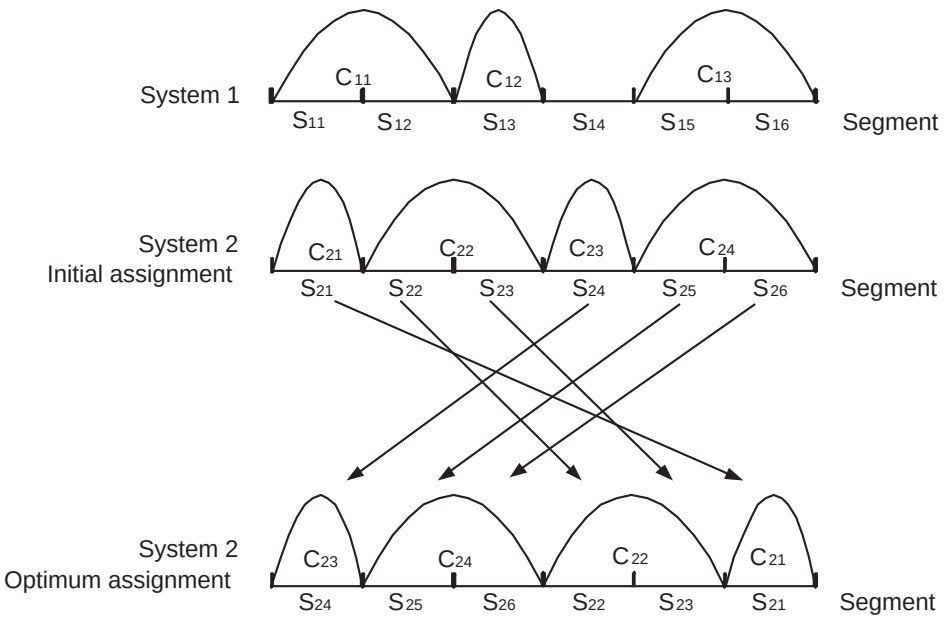
limiting factor.

7.2 Problem Description

Calculations of co-channel interference are necessary when solving the FAP in satellite systems. However, as indicated in [78], these calculations involve nonlinear terms in their mathematical expressions. In order to avoid treatment of nonlinearity, segmentation proposed by [78] divides the commonly shared frequency band into a number of segments. The carrier assignment is then made by placing the carriers at the integral multiples of unit segment. Through the segmentation, the nonlinear terms in the calculation of interference can be evaluated in a linear manner.

Fig. 7.4 indicates a example of segmentation. Each carrier in both systems occupies different length of frequency, with large carriers occupying several segments and small carriers occupying only one segment. In each system, every carrier can be divided into one or more consecutive unit segments. Thus the interference be-

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(a)

		C11		C12		C14	
		S11	S12	S13	S14	S15	S16
C21	S21	20	20	40	0	25	25
C22	S22	50	10	30	0	55	*
	S23	*	50	30	0	15	55
C23	S24	30	30	45	0	35	35
	S25	45	5	25	0	50	*
C24	S26	*	45	25	0	10	50

(b)

Figure 7.4: Segmentation and interference matrix for the system in Fig. 7.3. (a) Segmentation of the two systems. (b) the interference matrix $E = (e_{ij})$ for the M-segment system, where the S_{xy} stands for the y-th segment in system x.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

tween the two systems (say M segments) in Fig. 7.3 is described by an $M \times M$ interference matrix $E = (e_{ij})$, where the symbol '*' represents infinity. The ijth element e_{ij} represents the co-channel interference when segment i in system 2 uses a common frequency with segment j in system 1. Fig. 7.4 (a) shows the segmentation of the system in Fig. 7.3 and the optimum assignment under the interference matrix in Fig. 7.4 (b). In order to reduce the co-channel interference between the two adjacent systems, the frequency assignments in system 2 are re-arranged but the assignments in system 1 are fixed. In this way, the FAP is equivalent to assigning each segment in system 2 to a segment in system 1 using a common frequency.

The objectives of the assignment problem in satellite communication systems are:

1) minimize the largest element of the interference matrix selected in the assignment.

2) minimize the sum of interference of all the selected elements.

The three constraints are:

1) every segment in system 2 must be assigned to a segment in system 1.

2) each segment in system 1 can assigned at most one segment in system 2.

3) all segments of each carrier in system 2 should be assigned to consecutive segments in system 1 in the same order.

7.3 Existing Techniques for the FAP

The frequency reassignment can be formulated as a combinatorial optimization problem known as the frequency assignment problem (FAP), which is proven to

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

be NP-complete [78]. Due to the NP-complete nature of this assignment problem, heuristic methods, especially neural networks, are commonly adopted in recent research. However, early efforts in [78] applied an exact method using the branch-and-bound method.

The distinguished work of Mizuike and Ito [78] makes the optimization of frequency assignment in satellite systems easier to deal with in mathematical form by introducing the segmentation of carriers. They used branch-and-bound to solve several practical problems, including both inter-system problems and intra-system problems. The branch and bound algorithm although can find the global optimal assignments for these problems; however it fails when applied to large instances as indicated in [38].

Funabiki and Nishikawa [38] used gradual neural network (GNN) which consists of $N \times M$ binary neurons for the N-carrier-M-segment system with gradual expansion scheme of activated neurons. They sorted the $N \times M$ neurons with the ascending order of interference and divided them into P groups $g_1, g_2, ..., g_P$, where g_1 is the group of neurons with the smallest interference and g_P is the group of neurons with the largest interference. The GNN starts with group g_1 and searches the solution in several phases, with each phase adding one next smallest group of neurons. The algorithm stops when one feasible solution is found or all neurons have been added. The objective of the FAP can be achieved by starting to search from the neurons with the smallest interference. However, the selection of parameter P is difficult when the problem size is large and different P should be tried for difference problems. Furthermore, the multi-phase searching inevitably leads to heavy computation, especially for large problems.

Salcedo-Sanz *et al.* [93] presented a hybrid method which combines the Hopfield network and simulated annealing (HopSA) to tackle the FAP. The Hopfield neural network manages the constraints and simulated annealing improves the solutions obtained. The hybrid method separates the goal functions from the constraints. It is more scalable than the previous method. However, the method

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

needs more computation to find optimal or near optimal solutions compared with other neural network methods, e.g., the GNN due to the nature of the simulated annealing technique.

In this Chapter, we propose a scalable algorithm called noisy chaotic neural network with variable threshold, which separates the objective term from the constraint terms. As we can see from the discussion in Chapter 3, different threshold of neuron determines its firing probability, i.e., different threshold determines the selection of neurons in the final solutions. Through mapping the objective of the problem into the threshold of neuron in the neural network, we can cut the objective term from the energy function formulation. The constraints terms left over are formulated in the energy functions. The NCNN-VT not only reduces the number of terms in the energy function, but also has a higher convergence rate and a faster computation speed compared with the hybrid method in [93]. In the next Section, the application of the proposed NCNN-VT will be stated in detail.

7.4 Application of the NCNN-VT to the FAP

7.4.1 Neural network formulation

In this Chapter, we use a two-dimensional neural network which consists of $N \times M$ neurons for the FAP of N carriers and M segments. The output of each neuron V_{ij} will be converted into binary values V_{ij}^d. V_{ij}^d represents whether carrier i is assigned to segment $j-(j+c_i-1), (i=1,...,N; j=1,...,M)$, where c_i indicates the length of carrier i, i.e.:

$$V_{ij}^d = \begin{cases} 1 & \text{carrier } i \text{ is assigned to segment } j-(j+c_i-1), \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 7.5 shows the neural network formulation for the 4-carrier-6-segment

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

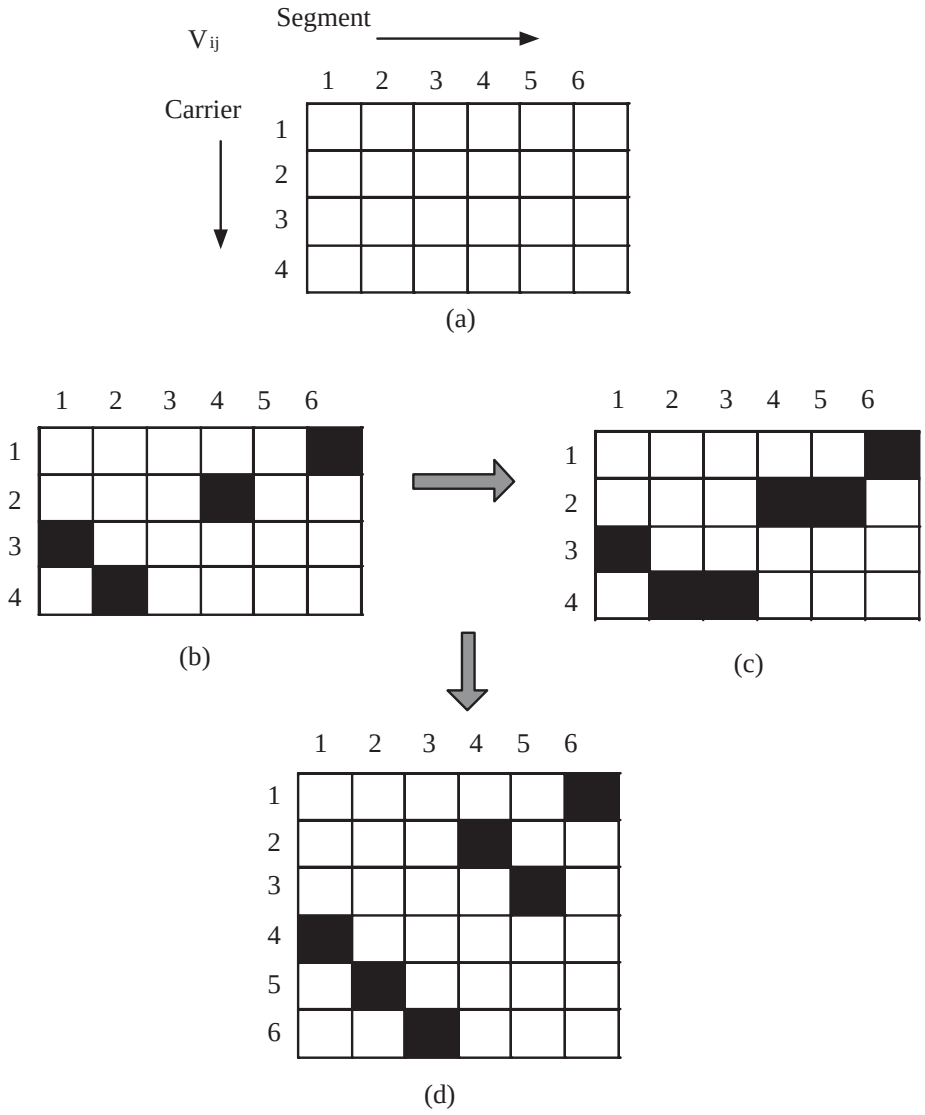


Figure 7.5: The neural network formulation for the FAP. (a) The 24 neurons for the 4-carrier-6-segment FAP. (b) the convergence state of the neural network. (c) the full assignment of the neural network formulation. (d) the final assignment of the segments for the FAP through the expansion from the neural network formulation.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

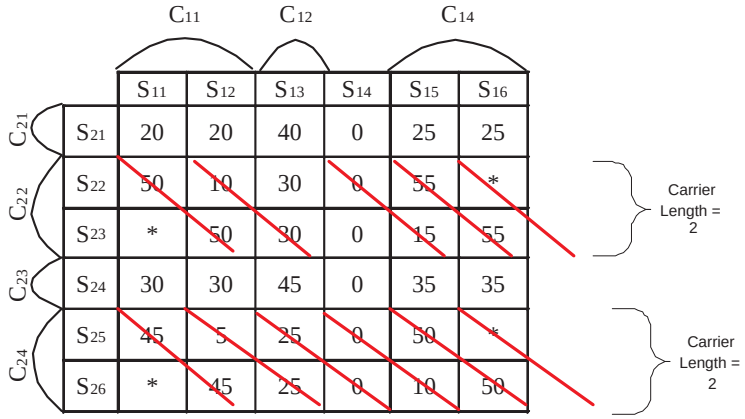
problem in Fig. 7.4. This neural network consists of 24 ($= 4 \times 6$) neurons as shown in Fig. 7.5 (a). Fig. 7.5 (b) is the convergence state, with the black squares stand for the neurons with output $V_{ij}^d = 1$. Fig. 7.5 (c) shows the full assignment for each carrier. And Fig. 7.5 (d) is the final frequency assignment for the FAP. Note that it can be easy to expand the convergence state in Fig. 7.5 (c) to the final assignment in Fig. 7.5 (d) given the carrier length for each carrier. We provide only the solution format in Fig. 7.5 (c) to represent the final assignment in this Chapter.

7.4.2 The Mapping Function

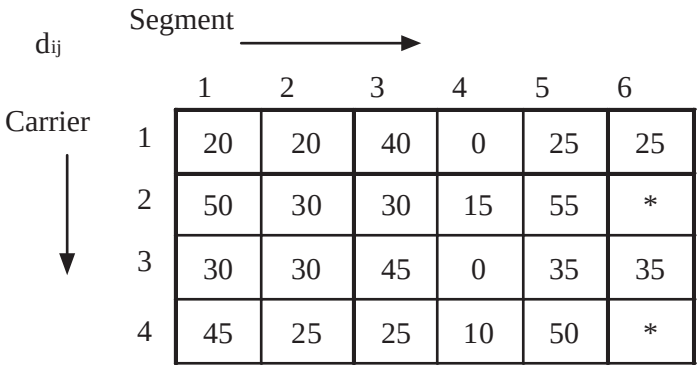
The cost of each neuron is problem-specific and can be in different forms. It is related to the goal function of the problem. In this FAP, the cost is the interference if neuron ij is firing, i.e., $V_{ij}^d = 1$. We label the cost matrix $D = (d_{ij}), i = 1, 2, ..., N; j = 1, 2, ..., M$. Cost d_{ij} for neuron ij is given by the largest element in interference matrix among c_i elements $e_{kj}, e_{k+1,j+1}, ..., e_{k+c_i-1,j+c_i-1}$, if we assign the carrier i to segments $j - (j + c_i - 1)$, where k is the first segment number of carrier i in the interference matrix and c_i is the length of carrier i [38]. Fig. 7.6 shows the way to compute the cost matrix from interference matrix E in Fig. 7.4 (b). If the carrier length for carrier i is 1, i.e., c_i=1, then line i in the cost matrix for carrier i is the same value as in the interference matrix for carrier i. If the carrier length is greater than 1, then we compute the largest value among the values marked by the diagonal line for each $j, (j = 1, 2, ...M)$ in Fig. 7.6.

Our objective is to minimize the largest element of the interference matrix selected in the assignment and at the same time minimize the sum of interference of all selected elements. Thus we define the choice of the mapping function of I_{ij}

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library



(a)



(b)

Figure 7.6: Cost computation for each neuron from interference matrix E. (a) The computation method, where the lines in (a) show the crossed elements which are needed in computing the maximum cost. (b) The final cost matrix obtained using the computation method.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

as follows:

$$I_{ij} = 1 - \frac{d_{ij} - d_{i,min}}{d_{i,max} - d_{i,min}}$$

$$= \frac{d_{i,max} - d_{ij}}{d_{i,max} - d_{i,min}} \,. \tag{7.1}$$

where d_{ij} is the ij-th element in cost matrix $D = (d_{ij}, i = 1, ..., N; j = 1, ..., M)$, and $d_{i,max}$ [1] is the maximum value in line i of matrix D and $d_{i,min}$ is the minimum value line i of matrix D. We will label $d_{i,max}$ as d_{max} and $d_{i,min}$ as d_{min} for simplicity.

Through the mapping in eqn. (7.1), not only can we achieve the objectives of FAP, but also separate the objective from the energy function, which will make the tuning of weighting coefficients in the energy function easier without the need to balance the optimization term and constraint term in one energy function. Moreover, it will improve the convergence speed of the noisy chaotic neural network as shown in the result discussion section.

7.4.3 Energy function

Recall that the goal of the FAP has been separated from the constraints. Only the two constraints of the FAP need to be formulated in the energy function. The first constraint of an N-carrier-M-segment problem is that each first segment of the N carriers in system 2 must be assigned to one and only one of the M segments. Hence, one and only one neuron among the M neurons for each carrier has output 1. Then the first constraint can be formulated as [38]:

$$E_1 = \sum_{i=1}^{N} (\sum_{q=1}^{M} V_{iq} - 1)^2 \,. \tag{7.2}$$

[1] Note that the maximum value of cost does not include the infinity value of in the interference matrix. Actually the neurons corresponding to the infinite interference will never fire due to its inhibitive cost.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

20	20	40	0	25	25
50	30	30	15	55	*
30	30	45	0	35	35
45	25	25	10	50	*

(a) Cost matrix

0.5	0.5	0	1	0.38	0.38
0.125	0.625	0.625	1	0	0
0.33	0.33	0	1	0.22	0.22
0.125	0.625	0.625	1	0	0

(b) Probability matrix of linear mapping

0.25	0.25	0	1	0.14	0.14
0.02	0.39	0.39	1	0	0
0.11	0.11	0	1	0.05	0.05
0.02	0.39	0.39	1	0	0

(c) Probability matrix of nonlinear mapping- case 1

0.71	0.71	0	1	0.61	0.61
0.35	0.79	0.79	1	0	0
0.58	0.58	0	1	0.47	0.47
0.35	0.79	0.79	1	0	0

(d) Probability matrix of nonlinear mapping - case 2

Figure 7.7: Probability matrix of neuron firing by different mapping functions (a) cost matrix. (b) probability matrix obtained by linear mapping in eqn. (7.1). (c) probability matrix obtained by nonlinear mapping in eqn. (3.4), and (d) probability matrix obtained by nonlinear mapping in eqn. (3.6).

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

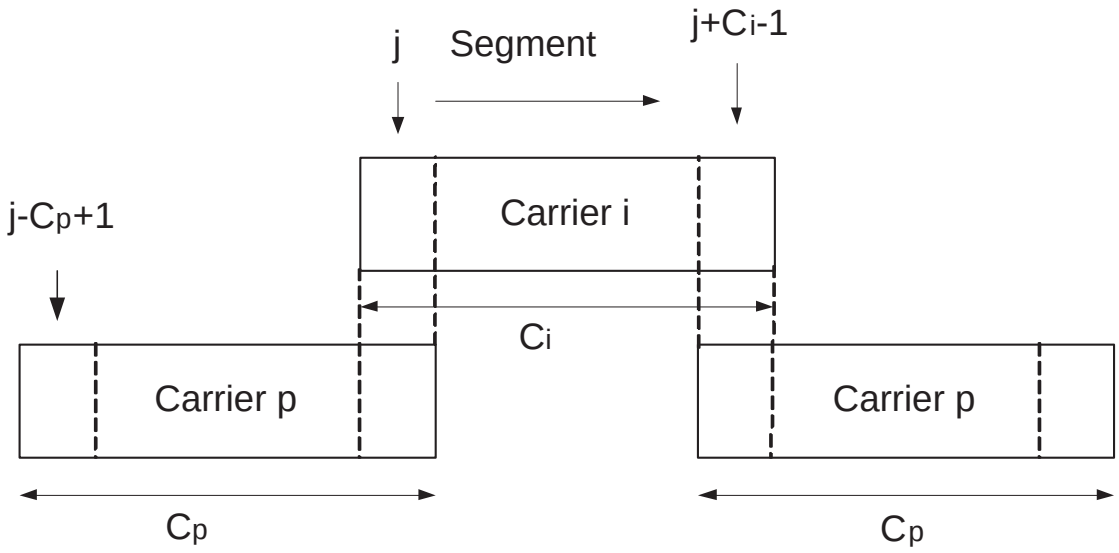


Figure 7.8: Violation condition for the second and third constraints of the FAP.

The second and third constraints are that each segment in system 1 can be assigned to at most one segment in system 2 and the assignment of carriers in system 2 should be in consecutive segments in system 1 in the same order. If carrier i is assigned to segment $j-(j+c_i-1)$, any other carrier $p(p \neq i)$ must not be assigned to the consecutive segment $(j-c_p+1)-(j+c_i-1))$. The violation condition of the two constraints is shown in Fig. 7.8, where c_i is the carrier length of carrier i. In other words, if carrier i is assigned with consecutive c_i segments, the segments occupied by any other carrier cannot occupy with the segment $j-(j+c_i-1)$. The first segment of each carrier $p, (p \neq i)$ should be $(j-c_p+1)$ before and $j-(j+c_i-1)$ after the first segment of carrier i.

Hence these two constraints give raise to the second part of the energy function to be minimized, as formulated in [38]:

$$E_2 = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{\substack{p=1 \\ p \neq i}}^{N} \sum_{q=j-c_p+1}^{j+c_i-1} V_{ij} V_{pq} \,. \tag{7.3}$$

Note that because $(j-c_p+1)$ can be negative and $(j+c_i-1)$ can exceed the total number of segments M. The original formulation in eqn. (7.3) [38] has

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

errors in dealing with the bounds and produces the program bugs when searching the solutions. We formulate the second term in our energy function in a revised version as follows:

$$E_2' = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{\substack{p=1\\p\neq i}}^{N}\sum_{q=max(j-c_p+1,1)}^{min(j+c_i-1,M)} V_{ij}V_{pq}\,. \tag{7.4}$$

where function $max(x, y)$ returns the maximum value between (x, y) two numbers and $min(x, y)$ finds the minimum value between (x, y).

We use the following convergence term in our energy function to help the neuron output converge to the corner (0 or 1) of the hypercube:

$$E_3 = \sum_{i=1}^{N}\sum_{j=1}^{M} V_{ij}(1 - V_{ij})\,, \tag{7.5}$$

The total energy function of the NCNN-VT is given by the summation of the three parts E_1, E_2', and E_3:

$$\begin{aligned} E = {} & \frac{W_1}{2}\sum_{i=1}^{N}(\sum_{q=1}^{M} V_{iq} - 1)^2 \\ & + \frac{W_2}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{\substack{p=1\\p\neq i}}^{N}\sum_{q=max(j-c_p+1,1)}^{min(j+c_i-1,M)} V_{ij}V_{pq} \\ & + \frac{W_3}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} V_{ij}(1 - V_{ij})\,. \end{aligned} \tag{7.6}$$

where W_1, W_2, and W_3 are weighting coefficients.

If not using the NCNN-VT but the NCNN, then the fourth part of the energy function should be added in the energy function, i.e., optimization term W_4 is needed to fulfill the objective of the FAP:

$$E_4 = W_4\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}V_{ij} \tag{7.7}$$

The choices of W_1, W_2, and W_3 are based on the rule that all terms in the energy function should be comparable in magnitude, so that none of the energy

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

terms dominates. The balance of each term in the energy function is crucial to the parameter selection. With the optimization term in the energy function, the tuning of weighting coefficients will be more difficult because the E_1 to E_3 are much smaller compared to E_4 [1], which makes the final solution very sensitive to the choice of W_4.

On the contrary, if we separate the optimization term from the constraint terms, it can be conclude that the balance between each term in the energy function and the selection of weighting coefficients will be simpler and easier, so do the selection of weighting coefficients.

From equations (2.13), (2.16) and (7.6), the dynamic equation for the NCNN-VT can be obtained:

$$y_{jk}(t+1) = k y_{jk}(t) + \alpha \left\{ -W_1\left(\sum_{q=1}^{M} V_{iq} - 1\right) - W_2\left(\sum_{\substack{p=1 \\ p\neq i}}^{N} \sum_{q=max(j-c_p+1,1)}^{min(j+c_i-1,M)} V_{pq}\right) - \frac{W_3}{2}(1-2V_{ij}) \right\} - z(t)\left[x_{jk}(t) - I_{jk}\right] + n(t)\,. \quad (7.8)$$

The neuron output is continuous between 0 and 1. We convert the continuous output V_{ij} of neuron ij to discrete neuron output V_{ij}^d as follows [125]:

$$V_{ij}^d = \begin{cases} 1 & \text{if } V_{ij} > \sum_{k=1}^{N}\sum_{l=1}^{M} V_{kl}(t)/(N\times M) \\ 0 & \text{otherwise} \end{cases}$$

The NCNN is updated cyclically and asynchronously. The new state information of a neuron is immediately available for the other neurons in the next

[1] The magnitude of E_4 is related with cost matrix D which is problem specific. It may vary in a big amount when the range of interference in the simulation varies from $1-100$ to $1-1000$.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

iteration. The iteration is terminated once a feasible assignment is obtained or the computation steps exceeds the predefined maximum number of iteration steps.

7.4.4 Computational Complexity of the NCNN-VT

We now analyze the computational complexity of the NCNN-VT. For an N-carrier-M-segment FAP, each neuron updates once in an iteration step. Hence, our algorithm has worst time complexity of $O(NM)$ in one iteration. However, it is difficult to determine the exact number of iterations required for different problem instances with different problem sizes. For large-size problems, the number of iterations needed for finding an feasible solution will be greater than in small-size problems. This is also true for the computation time. And we will include the computation time for each instances simulated in this work in the result discussion.

7.5 Computational Results and Comparisons

7.5.1 Parameter Discussion

The NCNN-VT contains two types of parameters, i.e., model parameters in the NCNN model and weighting coefficients in the energy function of the neural networks. We choose the model parameters based on the rule that the parameters should produce rich and flexible neuro-dynamics. The choices of these model parameters are similar to those used in other optimization problems [118] [116]. For example,

$$\begin{aligned} k = 0.9, \alpha = 0.015, \beta_1 = 0.001, \beta_2 = 0.0002 \\ \varepsilon = 1/250, I_0 = 0.65, z_0 = 0.08, A[n(0)] = 0.009\,. \end{aligned} \tag{7.9}$$

The selection of weighting coefficients is based on the rules that all terms in the energy function should be comparable in magnitude, so that none of them dominates. Note that the parameters listed here are empirical, some tuning on these

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

parameters is needed when applied to different problems, especially the weighting coefficients. We list the parameters for each problem solved in this chapter in Table 7.1.

7.5.2 Simulation Problems

We simulate our neural networks using desktop computers. We code the NCNN-VT in C languages and run in a Linux cluster with a parallel C/C++ compiler. The Linux cluster is a 16-node Dual Xeon 3.06 GHz (Intel IA32) cluster with applications, such as Matlab and parallel C/C++ programs. The cluster toolkit used is NPACI ROCKS v3.0.

In order to show the performance of our proposed method, we compare our results with previous methods in two types of examples. One type of examples consists of benchmark examples (BM 1 to BM 8) which were solved by the previous methods. The other type of examples (Case 9 to Case 20) consists of randomly generated large-size problems for the FAP. In order to generate the randomly instances, the number of carriers N and the number of segments M are chosen at first for each instance. Then the range of carrier length and the range of interference are determined. The value of carrier length for each carrier and the interference between the two systems will be generated as uniformly distributed random values. We generate three groups of random instances. Case 9 to Case 11 are generated in order to show the influence of the interference, by varying the interference while fixing the number of carriers, the number of segments, and the range of carrier length. Case 12 to Case 17 are intended to show the influence of carrier length. Case 18 to Case 20 are used to show the ability of the proposed method to deal with large numbers of carriers. The results of the benchmark examples show the capability of our method in solving the FAP. And the results of the large-size randomly generated problem demonstrate the scalability and applicability of our NCNN-VT.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 7.1: Parameters of the NCNN-VT model for the benchmark examples, with $W_1 = 1.0$.

Case	W_2	W_3	k	ϵ	I_0	α	$z(0)$	$n(0)$	β_1	β_2
1	1.0	0.7	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.001
2	1.0	0.7	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.001
3	0.3	0.7	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.001
4	0.3	0.7	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.001
5	0.4	0.7	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.0001
6	0.2	0.6	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.0001
7	0.2	0.6	0.9	1/250	0.65	0.015	0.08	0.01	0.001	0.0001
8	0.1	0.3	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.0001
9	0.2	0.4	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.001
10	0.2	0.6	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.0001
11	0.2	0.6	0.9	1/250	0.65	0.015	0.08	0.01	0.001	0.0001
12	0.2	0.4	0.9	1/250	0.65	0.015	0.08	0.01	0.001	0.0001
13	0.2	0.4	0.9	1/250	0.65	0.015	0.08	0.01	0.001	0.0001
14	0.1	0.4	0.9	1/250	0.65	0.015	0.08	0.01	0.001	0.0001
15	0.1	0.5	0.9	1/250	0.65	0.015	0.08	0.01	0.001	0.0001
16	0.1	0.5	0.9	1/250	0.65	0.015	0.08	0.01	0.001	0.0001
17	0.1	0.5	0.9	1/250	0.65	0.015	0.08	0.01	0.001	0.0001
18	0.1	0.6	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.0001
19	0.1	0.6	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.0001
20	0.2	0.4	0.9	1/250	0.65	0.015	0.08	0.02	0.001	0.001

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 7.2: Specifications of the benchmark FAP used in the simulations.

Instance	Number of carriers N	Number of segments M	Range of carrier length	Range of interference
BM 1	4	6	1 - 2	5 - 55
BM 2	4	6	1 - 2	1 - 9
BM 3	10	32	1 - 8	1 - 10
BM 4	10	32	1 - 8	1 - 100
BM 5	10	32	1 - 8	1 - 1000
BM 6	18	60	1 - 10	1 - 100
BM 7	30	100	1 - 10	1 - 100
BM 8	15	50	1 - 8	1 - 1000
Case 9	50	200	1 - 10	1 - 10
Case 10	50	200	1 - 10	1 - 100
Case 11	50	200	1 - 10	1 - 1000
Case 12	80	200	1 - 5	1 - 100
Case 13	80	300	1 - 8	1 - 100
Case 14	80	400	1 - 10	1 - 100
Case 15	80	450	1 - 12	1 - 100
Case 16	80	500	1 - 14	1 - 100
Case 17	80	600	1 - 16	1 - 100
Case 18	100	500	1 - 10	1 - 100
Case 19	150	400	1 - 5	1 - 100
Case 20	200	300	1 - 2	1 - 100

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

0 429 219 546 936 811 45 603 230 165 833 281 402 863 594 274 674 509 559 946 206 333 644 54 0 251 366 781 42 * * *
* 834 917 287 462 675 381 821 528 257 202 808 717 856 806 346 210 603 239 833 19 266 504 264 0 443 364 314 932 103 * *
* * 41 478 326 183 212 549 145 760 984 384 917 349 990 649 222 726 448 216 913 74 29 775 0 286 142 678 831 957 247 *
* * * 637 496 339 206 85 644 139 395 138 445 579 903 19 135 432 596 623 529 859 234 1000 0 550 166 287 444 851 830 316
0 265 709 452 119 560 574 704 783 224 143 80 867 95 736 112 836 73 291 217 887 893 794 764 0 806 534 177 23 283 639 *
* 344 705 655 68 179 808 577 357 476 425 502 175 106 636 30 409 273 354 392 352 888 87 100 0 401 607 988 313 502 399 590
0 518 493 663 190 654 492 98 652 319 729 33 343 175 609 958 663 545 193 378 257 563 730 866 0 490 379 982 300 * * *
* 22 362 288 321 122 959 548 482 150 185 67 772 925 729 279 483 167 774 237 811 47 911 959 0 686 123 523 883 661 * *
* * 560 965 28 429 565 793 141 686 452 458 499 170 318 885 970 398 204 572 223 142 526 721 0 691 524 665 633 614 635 *
* * * 325 192 998 296 953 803 85 782 86 904 334 702 869 178 681 708 921 468 422 464 593 0 449 892 252 422 82 500 208
0 566 307 797 168 637 298 159 996 733 905 441 361 926 388 426 145 938 311 765 33 340 113 192 0 62 885 40 865 * * *
* 245 303 663 678 153 819 695 207 952 118 349 616 268 735 596 293 231 487 463 827 613 628 709 0 915 971 458 294 288 * *
* * 213 311 937 504 45 199 513 629 761 19 164 979 206 157 273 503 242 268 603 9 557 98 0 250 474 248 646 446 485 *
* * * 48 247 663 281 353 798 195 308 468 136 358 503 611 287 917 142 949 638 677 865 566 0 418 760 376 878 804 494 792
0 960 425 677 605 590 164 877 226 518 925 654 577 145 980 932 404 328 901 155 56 171 898 370 0 * * * * * * *
* 313 955 139 999 338 372 785 323 117 942 752 573 598 71 80 810 65 355 689 791 144 666 713 0 806 * * * * * *
* * 236 947 298 771 965 925 528 760 982 734 339 359 37 552 795 223 675 857 925 835 461 351 0 776 15 * * * * *
* * * 151 886 637 682 593 702 984 69 234 844 554 482 175 674 124 445 701 491 725 221 564 0 819 312 579 * * * *
* * * * 984 861 640 245 942 991 596 977 765 744 730 744 884 272 932 576 433 968 911 5 0 404 872 332 377 * * *
* * * * * 3 587 582 514 90 756 828 187 16 816 522 210 662 230 397 464 936 688 851 0 817 193 767 348 475 * *
* * * * * * 375 947 508 920 282 762 117 652 187 924 585 28 902 656 381 473 996 574 0 3 31 642 657 117 792 *
* * * * * * * 949 512 62 335 56 419 17 899 626 185 825 506 71 102 603 246 565 0 937 587 96 920 868 606 192
0 706 481 737 465 648 221 24 586 707 831 394 391 257 727 815 777 589 567 274 392 220 721 760 0 542 383 342 323 271 644 310
0 65 507 945 601 851 160 114 676 629 862 830 306 729 215 311 901 263 274 25 196 586 135 338 0 341 691 488 413 632 410 *
* 158 278 191 272 42 32 884 790 167 785 939 572 726 710 896 905 109 133 290 883 951 823 511 0 196 149 916 546 384 734 910
0 979 749 742 163 231 37 547 976 623 280 606 449 916 902 412 241 852 300 512 155 915 637 875 0 866 933 365 180 516 730 *
* 542 413 341 543 563 418 242 261 307 471 114 801 124 334 456 950 399 994 659 143 51 125 311 0 194 844 951 241 779 798 843
0 354 791 436 383 881 99 328 144 211 961 847 620 703 750 425 723 594 220 495 471 98 300 538 0 657 317 221 736 329 963 250
0 404 632 418 441 375 964 542 506 101 608 58 757 886 391 694 52 254 626 436 962 90 488 740 0 278 957 114 98 456 * *
* 348 867 372 859 686 837 683 652 224 815 12 141 229 534 60 894 247 110 684 96 289 20 580 0 3 23 648 460 467 909 *
* * 426 842 461 185 275 285 253 929 102 466 452 697 351 558 255 36 189 203 371 190 528 725 0 227 716 721 493 189 114 960
0 236 179 133 321 964 406 115 871 696 364 415 120 485 473 302 830 719 582 517 5 605 714 952 0 215 182 349 75 611 803 740

Figure 7.9: The interference matrix and the best solution for BM 5.

Table 7.2 summarizes the main characteristics of the benchmark problems. The interference matrix and carrier length in the randomly generated instances are uniformly random numbers in their ranges. We run each instance 100 times with different initial conditions of neural networks. Fig. 7.9 shows the interference matrix for BM 5 and the best assignment achieved by the NCNN-VT for the BM 5 is displayed in Fig. 7.10. The best assignment is obtained in 30 different runs. The symbol " $*$ " in Fig. 7.9 stands for an infinity value of interference, and the black square in Fig. 7.10 represents the assignment.

7.5.3 Result Discussions

Table 7.3 shows the comparison of the best results obtained by the NCNN-VT and other previous methods. For the benchmark problems from BM 1 to BM 4, the NCNN-VT algorithm matches or improves the results of other existing algorithms. Our method can obtain better results when the problems become more complicated

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

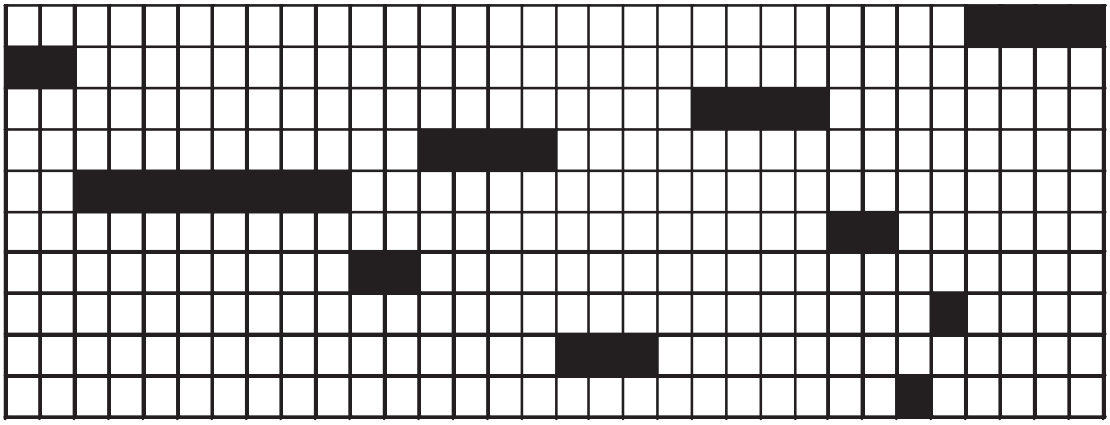


Figure 7.10: The best assignment achieved by the NCNN-VT for BM 5.

Table 7.3: Comparisons of the simulation results (largest interference and total interference) obtained by the NCNN-VT, GNN and HopSA in the benchmark examples.

Instance	GNN[38]		HopSA[93]		NCNN-VT	
	largest	total	largest	total	largest	total
BM 1	30	100	30	100	30	100
BM 2	4	13	4	13	4	13
BM 3	7	85	7	85	7	88
BM 4	64	880	84	886	64	880
BM 5	640	8693	817	6851	**640**	**7246**
BM 6	49	1218	48	1002	**35**	**1029**
BM 7	100	4633	98	4093	**61**	**2824**
BM 8	919	16192	891	14495	**695**	**15373**

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 7.4: Comparisons of the NCNN and the NCNN-VT on the benchmark examples with 50 different runs, where η is the convergence rate, T is the average number of interation steps, I_{max} is the largest interference and I_t is the total interference. Here the interference is shown in best and average value (Best/Ave).

	NCNN				NCNN-VT			
	η (%)	T	I_{max} (Best/Ave)	I_t (Best/Ave)	η (%)	T	I_{max} (Best/Ave)	I_t (Best/Ave)
1	100	393.2	30/32.0	100/100.0	100	378.3	30/30.0	100/104.1
2	100	766.4	6/6.8	17/18.2	100	478.1	4/4.0	13/13.7
3	100	2241.7	8/9.80	101/108.6	100	43.6	7/7.0	88/92.2
4	100	2042.6	78/93.8	1025/1204.4	96	2157.5	64/65.1	880/895.6
5	92	1994.3	674/945.1	9279/11612.9	98	1819.8	640/677.0	7246/8613.8
6	80	2114.8	47/49.1	1057/1230.9	93	2415.9	35/42.8	1052/1074.9
7	85	3225.0	88/95.2	3173/3501.2	86	2890.7	61/79.9	2824/3008.7
8	70	8021.4	800/835.4	16321/17342.3	76	7769.0	695/877.7	15373/18033.6

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 7.5: Simulation results (largest interference and total interference) on the randomly generated instances between the HopSA [93] and the NCNN-VT. The results are obtained in 50 runs and are displayed as mean±SD (standard deviation).

Instance	HopSA		NCNN-VT	
	largest	total	largest	total
9	10.50±0.6	658.32±15.34	9.80±0.4	627.33±12.93
10	124.26±10.22	6533.19±123.52	90.93±5.78	5374±60.44
11	989.20±73.49	49982.36±150.51	910.91±64.42	49387.72±120.23
12	N/A	N/A	75.80±2.53	3437.30±74.25
13	N/A	N/A	76.03±11.37	6294.62±151.81
14	N/A	N/A	97.24±0.75	10826.91±351.21
15	N/A	N/A	92.30±6.43	12644.2±273.7
16	N/A	N/A	87.00±8.35	14385.43±283.83
17	N/A	N/A	99.67±0.71	21894.33±776.36
18	N/A	N/A	96.86±3.08	14841.29±628.95
19	N/A	N/A	98.75±1.67	9924.13±387.91
20	N/A	N/A	42.95±22.28	14487.65±434.26

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Table 7.6: Iteration steps, computation time (second), and convergence rate for the NCNN-VB on a Linux cluster with 16-node Dual Xeon 3.06 GHz (Intel IA32). The results are obtained through 50 runs and are displayed as mean±SD (standard deviation). Here convergence rate is defined as the ratio of the runs in which feasible solutions can be found to all number of runs.

	HopSA		NCNN-VT	
Instance	Time (Sec.) mean±SD	Convergence Rate	Time (Sec.) mean±SD	Convergence Rate
BM 1	1.0±0.0	100%	0.02±0.1	100%
BM 2	1.0±0.0	100%	0.02±0.1	100%
BM 3	33.0±0.0	100%	0.01±0.0	100%
BM 4	33.8±0.8	100%	2.1±0.4	96%
BM 5	33.0±0.5	100%	1.8±0.4	98%
BM 6	1529.9±156.1	100%	11.8±0.6	82%
BM 7	2845.0±528.2	100%	78.3±13.3	86%
BM 8	3738.1±614.7	75%	143.2±12.5	82%
Case 9	5389.1±823.2	80%	133.6±2.3	86%
Case 10	8916.3±1529.2	70%	172.7±5.8	100%
Case 11	15543.8±2116.6	70%	187.5±7.3	100%
Case 12	N/A	N/A	271.8±27.1	100%
Case 13	N/A	N/A	1034.0±56.1	93%
Case 14	N/A	N/A	1654.7±207.2	91%
Case 15	N/A	N/A	1851.7±353.9	100%
Case 16	N/A	N/A	1959.7±118.1	79%
Case 17	N/A	N/A	2621.0±118.4	78%
Case 18	N/A	N/A	2989.6±200.7	80%
Case 19	N/A	N/A	3550.2±233.3	85%
Case 20	N/A	N/A	2477.6±14.0	100%

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

as shown in bold face for BM 5 to BM 8 in Table 7.3. The results on the benchmark examples (BM 1 to BM 8) show that the NCNN-VT and the HopSA can find better solution than the GNN.

Paired t-test is performed between the HopSA and our NCNN-VT from 10 runs on case 9. The t value is 15.6. Entering a t table at 9 degrees of freedom (the number of replicate $n-1$) we find a tabulated t value of 2.26 ($p = 0.05$) going up to a tabulated value of 4.78 ($p = 0.001$). Our calculated t value exceeds these, so the difference between the two compared methods is very highly significant.

Table 7.4 shows the comparisons between the NCNN and the NCNN-VT on 8 benchmark problems. The results are obtained from 50 different runs. From this table, it can be seen that the NCNN-VT needs less number of interations than the NCNN does. Also the NCNN-VT can achieve better soultions with less largest and total interference. As to the convergence rate, both the NCNN and the NCNN-VT have high convergence rates in small cases, but have lower rates (down to 70%) when applied to large ones.

In order to compare the performance with the HopSA, we compared the results of the HopSA and NCNN-VT on the randomly generated instances from Case 9 to Case 20. We simulate the HopSA algorithm under the same simulation environment as the NCNN-VT. The results are summarized in Table 7.5, which shows that the HopSA can find solutions in the 50-node 200-segment problem in several days of computation. The HopSA method fails to find solutions in instances with larger sizes due to its excessive computation time. "N/A" in Table 7.5 represents the situations where the HopSA cannot find solutions in 1 month. Compared with our proposed NCNN-VT, the computation time needed to find a feasible solution takes several hours of CPU time in all instances.

We listed the iteration steps and computation time of our NCNN-VT for all the problems in Table 7.6 together with the convergence rate of our method. Here one iteration step in the NCNN-VT stands for one loop that all neurons are updated

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

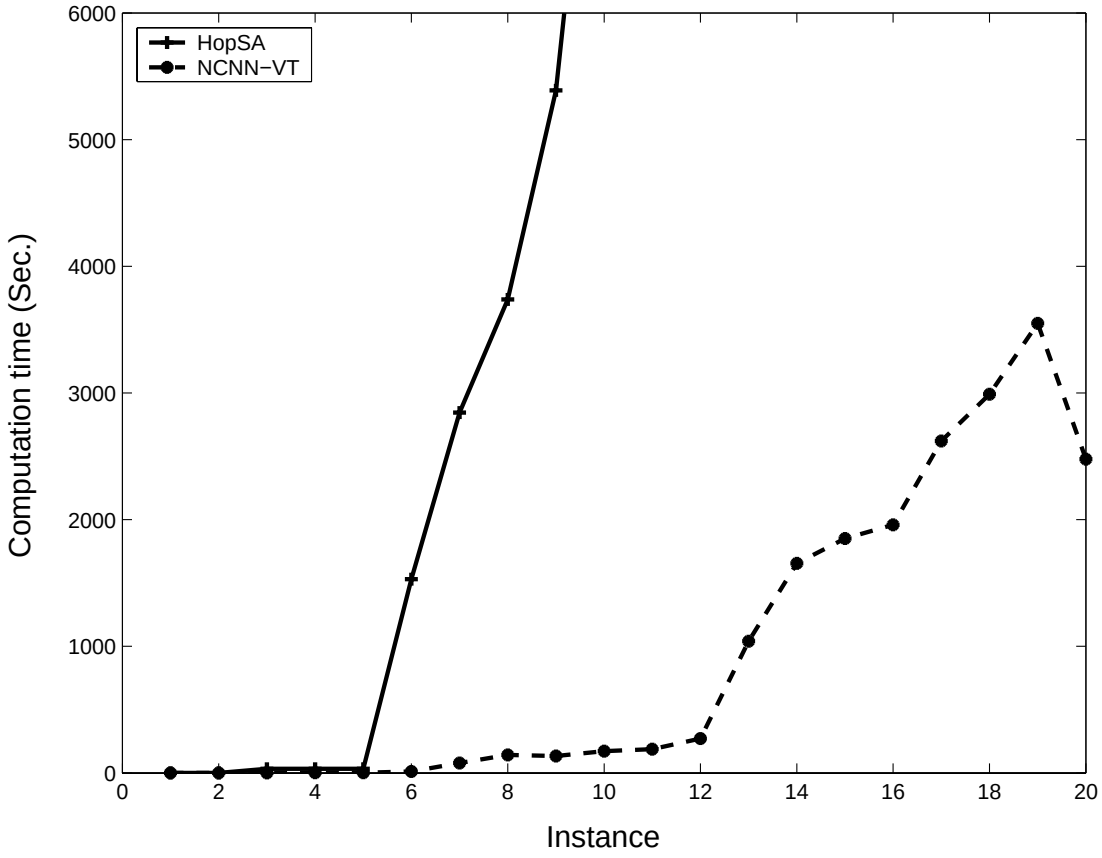


Figure 7.11: The computation time required by the NCNN-VT for the test instances. The x-axis represents the instances from 1 to 20, and the y-axis is the computation time required by each instance.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

once. And the computation time is measured in seconds (Sec.). From Table 7.6 we can see that our NCNN-VT achieves at lease 78% convergence rate in all instances. Furthermore, it can converge to a solution in smaller iteration steps and computation time compared with other methods. We plot the average computation time of 50 runs from Table 7.6 in Fig. 7.11, which shows that the computation time increases more slowly with the growing of the problem size compared to the HopSA. Compared with the HopSA which needs days for instances 9 to 11, the NCNN-VT needs not more than 200 seconds.

7.6 Concluding remarks

In this chapter, we have proposed a novel method called the noisy chaotic neural network with variable threshold and applied it to solve the frequency assignment problem in satellite communication systems. The NCNN-VT consisted of $N \times M$ neurons for an N-carrier-M-segment system with adaptive firing neurons. We used the linear mapping of parameter I_i to separate the objectives from the constraints in the FAP, thus simplified the formulation of the FAP in the energy function and made the tuning of weighting coefficients easier. The adaptive mapping scheme achieves the objectives of the FAP, and the noisy chaotic neural network aims to find a feasible solution satisfying the constraints through its complex neurodynamics with stochastic noise and flexible chaos. The simulation results of 20 benchmark instances showed that the NCNN-VT can find better solutions of co-channel interference compared to the previous methods with very low computational cost.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Chapter 8

Conclusion and Recommendations

In this chapter, we draw the conclusions of the thesis. The contents of the previous chapters are reviewed. Some recommendations for future work are provided.

8.1 Conclusions

In this thesis, we have focused on discussions of hybrid neural network methods and noisy chaotic neural network techniques with their applications on solving different COPs. The neural network approach is a promising technique to solve NP-hard combinatorial optimization problems. Hopfield Neural networks are potentially powerful heuristics for solving these problems because they are intrinsically parallel systems with significant potential for fast hardware implementation. With the chaotic simulated annealing, the chaotic neural networks have the ability to escape from the local minima and perform effective searching in solution space. By adding the decaying noise into the transiently chaotic neural network, our proposed NCNN has the ability of effective searching with deterministic chaotic dynamics and stochastic wandering in solution space.

In order to show the potential of the NCNN, we have applied it to solve the

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

TOP in network design with reliability constraint in Chapter 4. The TOP is a NP-hard COP with great importance in network design. With simulations to various instances of TOP with up to 300-nodes and 1501-edges, the NCNN shows the better performance than the previous methods and the ability to solve problem with large size.

Besides the application of the original NCNN model, we have proposed several original neural network methods in this thesis. Specifically, in Chapter 5, we have proposed a hybrid algorithm which combined sequential vertex coloring and the noisy chaotic neural network to solve the broadcast scheduling problem in wireless multihop networks. We used two stages for the two objectives of the BSP, i.e., we used sequential vertex coloring to find the minimal TDMA frame length with transmission scheduling in the first stage and the NCNN to maximize channel utilization and minimize time delays in the second stage. We evaluated our hybrid algorithm in three benchmark examples plus three larger instances. The results showed that our hybrid method finds better solutions than the other algorithms did in the examples.

An application of our proposed gradual noisy chaotic neural network is discussed in Chapter 6. We proposed a gradual noisy chaotic neural network model for solving the broadcast scheduling problem (BSP) in packet radio networks. A two-phase optimization have been adopted to solve the two objectives of the BSP with two different energy functions. The G-NCNN is used to find an optimal transmission schedule with the minimal TDMA frame length in the first phase. In the second phase, additional node transmissions are found using the NCNN based on the results from the first phase. We evaluate our G-NCNN algorithm in three benchmark examples and 600 randomly generated geometric graph instances. We compare our results with existing methods including mean filed annealing, hybrid method of HNN and GA, the sequential vertex coloring algorithm, and the gradually neural network. The results of the benchmark problems show that the G-NCNN always finds the best solutions with minimal average time delays and

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

maximal channel utilization among the existing methods. We shows that our G-NCNN has significant improvements over the HNN-GA through a paired t-test. We also compared the NCNN with the TCNN in three instances and showed that the NCNN has better performance than the TCNN.

A novel method called the noisy chaotic neural network with variable threshold is proposed in Chapter 8 and applied to solve the frequency assignment problem in satellite communication systems. The NCNN-VT consists of $N \times M$ neurons for the N-carrier-M-segment system with adaptive firing neurons. We use the linear mapping of parameter I_i to separate the objectives from the constraints in FAP, thus simplified the formulation of FAP in energy function and makes the tuning of weighting coefficients easier. The adaptive firing neuron aims to achieve the objectives of FAP, and the neural network with complex neuron dynamics with stochastic noise and flexible chaos is used to find a feasible solution. The simulation results of 20 instances show that the NCNN-VT can find better solution of cochannel interference than the previous methods with very low computational cost. The computation time needed for the NCNN-VT grows linearly with the problem size increasing. That means our NCNN-VT method is a highly scalable method for solving the optimization problems.

In summary, the application of the NCNN on the TOP, together with the results for other optimization problems in [118, 76, 113, 116] support the conclusion that the NCNN is an efficient approach for solving large-scale combinatorial optimization problems. Although the NCNN is a general method, it cannot be applied to the BSP without the gradual expansion scheme due to the first objective of the BSP (minimal number of time slot), which means not every COP can be formulated and solved by the NCNN model. By adopting the GES scheme, the G-NCNN has the potential of effective searching together with a flexible network structure by gradually span of its number of neurons. The extensive simulation results on the BSP shows the performance of the G-NCNN and demonstrate its potential in solving combinatorial optimization problems. However, the multi-phase

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

expansion of the G-NCNN may need the long computation time in searching good solutions, especially when the solution is found in the last stage. Based on the above observation, we proposed another original neural network model by combing the adaptive mapping scheme with the NCNN, i.e., the noisy chaotic neural network with variable threshold. The NCNN-VT has been applied to solve the FAP in satellite system. The simulation results show that the NCNN-VT can find better solution than previous methods with relatively less computation time.

8.2 Recommendations for Future Research

Although we have solved three practical combinatorial optimization in communications networks using the NCNN, the G-NCNN, and the NCNN-VT, there are still many other problems to be solved in order to demonstrate their potential. We have simplified the neural energy function formulation by separating the objectives from the constraints of the problems. But the selection of weighting coefficients in the energy function still needs to be selected by trial and error. Future research can be conducted on the following aspects:

1. Besides the COPs in communication networks, We should apply the proposed neural network model to solve other optimization problems, whether combinatorial or non-combinatorial optimization problems, whether in communications networks or in other industry and research areas.

2. It is still necessary to choose the parameters by trial and error in order to avoid invalid or poor solutions in our neural computation. An effective way to automatically or adaptively select the weighting coefficients without cause any more computation or complexity should be developed. An ideal situation is that the neural network can perform the searching whilst the adaptive tuning of its parameters. In other words, after the formulation of the problem into a energy function, the ideal neural network model can evolve

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

itself from the learning and interaction with the problem, and adaptively select its parameters.

3. There are requirements to use other kind of hybrid methods to tackle different problems, e.g., the NCNN combines with the GA and the SA. The investigation into these kind of hybridization maybe provides more efficient methods in solving COPs.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Author's Publications

1. Lipo Wang, Haixiang Shi, A Gradual Noisy Chaotic Neural Network for Solving the Broadcast Scheduling Problem in Packet Radio Networks". pp. 989-1000, Vol. 17, Issue 4, IEEE Trans. Neural Networks, July 2006.

2. Haixiang Shi, Lipo Wang, "Broadcast scheduling in wireless multihop networks using a neural-network-based hybrid algorithm", pp. 765-771, Vol. 18, Issues 5-6, Neural Networks.

3. Lipo Wang, Haixiang Shi, and Wen Liu, "Solving Frequency Assignment problem in Satellite Communications Systems using Noisy Chaotic Neural Network with Variable Threshold." Submitted to IEEE Transactions on Systems, Man, and Cybernetics, Part C.

4. Haixiang Shi, Lipo Wang, and Jacek M. Zurada, "Reliability Constraint Network Topological Optimization in Communication Networks Using a Noisy Chaotic Neural Network." Submitted to IEEE Transactions on Circuits and Systems I.

5. Wen Liu, Lipo Wang, and Haixiang Shi, "Transient chaos in cellular neural network." Submitted to IEEE Transactions on Circuits and Systems II.

6. Haixiang Shi, Lipo Wang, "An Adaptive Noisy Chaotic Neural Network Approach for Frequency Assignment in Satellite Communications Systems", Proceedings of ICONIP 2005, Taiwan, China.

7. Haixiang Shi, Lipo Wang, "Optimal TDMA Frame Scheduling in Broad-

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

casting Packet Radio Networks Using a Gradual Noisy Chaotic Neural Network. ICNC 2005: 1080-1089, ChangSha, China.

8. Haixiang Shi, Lipo Wang, "A Hybrid Neural Network for Optimal TDMA Transmission Scheduling in Packet Radio Networks". Proceedings of the International Joint Conference on Neural Networks, July 13 - August 4, Montreal, Canada, 2005.

9. Lipo Wang, Haixiang Shi, "A Noisy Chaotic Neural Network Approach to Topological Optimization of a Communication Network with Reliability Constraints". Proceedings of International Symposium on Neural Networks, Dalian, China, Part II, pp. 230 - 235, August 19-21, 2004.

10. Haixiang Shi and Lipo Wang, "A noisy chaotic neural network approach for broadcast scheduling problem in packet radio networks". The Second International Conference on computational Intelligence, Robotics and Autonomous Systems, CIRAS 2003

11. Haixiang Shi and Lipo Wang, "A mixed branch-and-bound and neural network approach for the broadcast scheduling problem". Design and Application of Hybrid Intelligent Systems, HIS03, the Third International Conference on Hybrid Intelligent Systems, Melbourne, Australia, December 14-1, pp. 42-49. 2003.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

Bibliography

[1] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines*. John Wiley, Chichester, 1989.

[2] S. Ablameyko, M. Gori, L. Goras, and V. Piuri, editors. *Energy-based computation with symmetric Hopfield nets*, volume 186. Amsterdam: IOS Press, 2003.

[3] H. M. AboElFotoh and L. S. Al-Sumait. A neural approach to topological optimization of communication networks, with reliability constraints. *IEEE Trans. Reliability*, 50:397–408, 2001.

[4] H. M. AboElFotoh and C. Colbourn. Computing 2-terminal reliability for radio-broadcast networks. *IEEE Trans. Reliability*, 38:538–555, 1989.

[5] H. M. AboElFotoh, S. Iyengar, and K. Chakrabarty. Computing reliability and message delay for cooperative wireless distributed sensor networks subject to random failures. *IEEE Trans. Reliability*, 54:145 – 155, 2005.

[6] C. W. Ahn and R. S. Ramakrishna. Qos provisioning dynamic connection-admission control for multimedia wireless networks using a Hopfield neural network. *IEEE Trans. Vehicular Technology*, 53:106–117, 2004.

[7] K. Aihara, T. Takabe, and M. Toyoda. Chaotic neural networks. *Physics Letters A*, 144:333–340, 1990.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[8] Y. Akiyama, A. Yamashira, M. Kajiura, Y. Anzai, and H. Aiso. The gaussian machine: a stochastic neural network for solving assingment problems. *Neural Network Comput.*, 2:43–51, 1991.

[9] M. K. M. Ali and F. Kamoun. Neural networks for shortest path computation and routing in computer networks. *IEEE Trans. on Neural Networks*, 4:9, 1993.

[10] H. I. amd K. Sekine and K. Imai. Computational investigations of all-terminal network reliability via bdds. *IEICE Trans. Fundamentals*, E82-A(5):714–721, 1999.

[11] M. Ball and R. M. V. Slyke. Backtracking algorithms for network reliability analysis. *Annals of Discrete Mathematics*, 1:49–64, 1977.

[12] C. M. Barnhart, J. E. Wieselthier, and A. Ephremides. A neural networks approach to solving the link activation problem in multihop radio networks. *IEEE Trans. Communications*, 43:1277–1283, 1995.

[13] D. Bertsekas and R. Gallager. *Data networks*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[14] S. Bhardwaj and Jayadeva. Sequential chaotic annealing neural network for cdma multiuser detection. In *Neural Information Processing, 2002. ICONIP '02*, volume 5, pages 2176–2180, 2002.

[15] R. D. Brandt, Y. Wang, A. J. Laub, and S. K. Mitra. Alternative network for solving the travelling salesman problem and the list-matching problem. In *Proceedings IEEE International Joint Conference on Neural Networks*, volume 2, pages 333–340, 1988.

[16] H. Cancela and M. E. Khadiri. Series-parallel reductions in monte carlo network-reliabilityevaluation. *IEEE Transactions on Reliability*, 47:159–164, 1998.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[17] H. Cancela and M. E. Urquhart. Adapting rvr simulation techniques for residual connectednessnetwork reliability models. *IEEE Transactions on Computers*, 51:439–443, 2002.

[18] J. Carlier and C. Lucet. A decomposition algorithm for network reliability evaluation. *Discrete Applied Mathematics*, 65:141–156, 1996.

[19] S. Cavalieri and O. Mirabella. Neural networks for process scheduling in real-time communication system. *IEEE Trans. Neural Networks*, 7:1272–1285, 1996.

[20] G. Chakraborty and Y. Hirano. Genetic algorithm for broadcast scheduling in packet radio networks. In *IEEE World Congress on Computational Intelligence*, pages 183–188, 1998.

[21] L. Chen and K. Aihara. Chaotic simulated annealing by a neural network model with transient chaos. *Neural Networks*, 8:915–930, 1995.

[22] L. Chen and K. Aihara. Chaos and asymptotical stability in discrete-time neural networks. *Physica D*, 104:286–325, 1997.

[23] L. Chen and K. Aihara. Global searching ability of chaotic neural networks. *IEEE Trans. Circuits and Systems - I: Fundamental Theory and Applications*, 46(8):974–993, 1999.

[24] L. Chen and K. Aihara. Strange atractors in chaotic neural networks. *IEEE Trans. Circuits and Systems - I: Fundamental Theory and Applications*, 47:1455–1468, 2000.

[25] N. Christofides. *Graph Theory - An Algorithmic Approach*. New York: Academic Press Inc., 1975.

[26] R. M. Closkey, A. Packard, and J. Sipila. Branch and bound computation of the minimum norm of a linear fractional transformation over a structured set. *IEEE Trans. on Automatic Control*, 45:369–375, 2000.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[27] C. Colbourn. *The combinatorics of network reliability.* Oxford Univ. Press, 1987.

[28] G. W. Davis. Sensitivity analysis in neural net solutions. *IEEE Trans. on Systems, Man and Cybernetics*, 19:1078–1082, 1989.

[29] D. Deeter and A. E. Smith. Heuristic optimization of network design considering all-terminal reliability. In *Proceedings Annual Reliability and Maintainability Symposium*, pages 194–199, 1997.

[30] D. E. V. den Bout and T. K. Miller. A traveling salesman objective function that works. In *Proceedings IEEE International Joint Conference on Neural Networks*, volume 2, pages 299–303, 1988.

[31] B. Dengiz, F. Altiparmak, and A. Smith. Efficient optimization of all-terminal reliability networks using an evolutionary approach. *IEEE Trans. Reliability*, 46:18–26, 1997.

[32] B. Dengiz, F. Altiparmak, and A. Smith. Local search genetic algorithm for optimal design of reliable networks. *IEEE Trans. Evolutionary Computational*, 1:179–188, 1997.

[33] R. Ebendt, W. Gunther, and R. Drechsler. An improved branch and bound algorithm for exact bdd minimization. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 22:1657–1663, 2003.

[34] A. Ephremides and T. V. Truong. Scheduling broadcast in multihop radio networks. *IEEE Trans. on Communications*, 38:456–460, 1990.

[35] F. Fogelman, E. Goles, and G. Weisbuch. Transient length in sequential iterations of threshold functions. *Discr. Appl. Math.*, 6:95–98, 1983.

[36] N. Funabiki and J. Kitamichi. A gradual neural network algorithm for broadcast scheduling problems in packet radio networks. *IEICE Trans. Fundamentals*, E82-A:815–824, 1999.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[37] N. Funabiki and S. Nishikawa. A binary hopfield neural-network approach for satellite broadcast scheduling problems. *IEEE Trans. on Neural Networks*, 8:441–445, 1997.

[38] N. Funabiki and S. Nishikawa. A gradual neural-network approach for frequency assignment in satellite communication systems. *IEEE Trans. Neural Networks*, 8:1359–1370, 1997.

[39] N. Funabiki and Y. Takefuji. A parallel algorithm for broadcast scheduling problems in packet radio networks. *IEEE Trans. on Communications*, 41:828–831, 1993.

[40] M. R. Garey and D. S. Johnson. *Computers and intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York, 1979.

[41] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Analysis and machine Intelligence*, 6:721–741, 1984.

[42] F. Glover. Tabu search i. *ORSA Journal on computing*, 1:190–206, 1989.

[43] F. Glover. Tabu search ii. *ORSA Journal on computing*, 2:4–32, 1990.

[44] E. Goles, F. Fogelman, and D. Pellegrin. Decreasing energy functions as a tool for studying threshold networks. *Discr. Appl. Math.*, 12:261–277, 1985.

[45] M. Hasegawa, T. Ikeguchi, and K. Aihara. Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problems. *Phys. Rev. Lett.*, 79:2344–2347, 1997.

[46] M. Hasegawa, T. Ikeguchi, and K. Aihara. Solving large scale traveling salesman problems by chaotic neurodynamics. *Neural Networks*, 15:271–283, 2002.

[47] Y. Hayakawa, M. A, and Y. Sawada. Effects of the chaotic noise on the performance of a neural network model for optimization problems. *Physical Review E*, 51:2693–2696, 2002.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[48] Y. He. Chaotic simulated annealing with decaying chaotic noise. *IEEE Trans. Neural Networks*, 13:1526–1531, 2002.

[49] S. Hegde, J. Sweet, and W. Levy. Determination of parameters in a hopfield/tank computational network. In *Proceedings IEEE International Conference in Neural Networks*, volume 2, pages 291–298, 1988.

[50] J. J. Hopfield. Neural networks and physical systems with emergent collective computional abilities. In *Proc. Natl. Acad. Sci. USA*, volume 79, pages 2554–2558, 1982.

[51] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. In *Proc. Natl. Acad. Sci. USA*, volume 81, pages 3088–3092, 1984.

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

[53] Y. Horio and K. Suyama. Experimental verication of signal transmission using synchronized sc chaotic neural networks. *IEEE Trans. Circuits and Systems - I: Fundamental Theory and Applications*, 42:393–395, 1995.

[54] C. Jaeseok, A. El-Keib, and T. Tran. A fuzzy branch and bound-based transmission system expansion planning for the highest satisfaction level of the decision maker. *IEEE Transactions on Power Systems*, 20:476–484, 2005.

[55] R. Jan and F. Hwang. Topological optimization problem of communication networks subject to a reliability constraint. volume 2, pages 487–494, 1990.

[56] R. Jan, F. Hwang, and S. Chen. Topological optimization of a communication network subject to a reliability constraint. *IEEE Trans. Reliability*, 42:63–79, 1993.

[57] Jayadeva. Sequential chaotic annealing and its application to multilayer channel routing. In *VLSI Design, 1999. Proceedings. Twelfth International Conference On*, pages 570–573, 1999.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[58] Jayadeva, S. D. Roy, and A. Chaudhary. The compact analog neural network-model of a new paradigm for neural based optimization, and its hardware realization. In *TENCON '98. 1998 IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control*, volume 1, pages 25–28, 1998.

[59] M. Jeruchim. A survey of interference problems and applications to geostationary satellite networks. In *Proceedings IEEE*, pages 317–331, 1977.

[60] L. Jie, K. Pattipati, P. Willett, and G. Levchuk. Fast optimal and suboptimal any-time algorithms for cdma multiuser detection based on branch and bound. *IEEE Trans. on Communications*, 52:632–642, 2004.

[61] D. Johnson, C. Papadimitriou, and M. Yannakakis. Optimzation by simulated annealing: an experimental evalution, part 1, graph partitioning. *Operat. Res.*, 37:865–892, 1989.

[62] D. Johnson, C. Papadimitriou, and M. Yannakakis. Optimzation by simulated annealing: an experimental evalution, part 2, graph partitioning. *Operat. Res.*, 39:378–406, 1991.

[63] D. Jungnickel. *Graphs, Netwrks and Algorithms*. Springer-Verlag, Berlin, Germany, 1999.

[64] B. Kamgar-Parsi and B. Kamgar-Parsi. Dynamical stability and parameter selection in neural optimization. In *Proceedings IEEE International Joint Conference on Neural Networks*, volume 4, pages 566–571, 1992.

[65] S. Kirkpatrick, C. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[66] T. Koide, S. Shinmori, and H. Ishii. Topological optimization with a reliability constraint. *Discrete Applied Mathematics*, 115:135–149, 2001.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[67] J. Kuri, N. Puech, M. Gagnaire, E. Dotaro, and R. Douville. Routing and wavelength assignment of scheduled lightpath demands. *IEEE Journal on Selected Areas in Communications*, 21:1231–1240, 2003.

[68] T. Kwok and K. Smith. A unified framework for chaotic neural-network approaches to combinatorial optimization. *IEEE Transactions on Neural Networks*, 10:978–981, 1999.

[69] T. Kwok and K. Smith. Experiment analysis of chaotic neural network models for combinatorial optimization under a unifying framework. *Neural Networks*, 13:731–744, 2000.

[70] T. Kwok and K. Smith. A noisy self-organizing neural network with bifurcation dynamics for combinatorial optimization. *IEEE Trans. on Neural Networks*, 15:84–98, 2004.

[71] W. K. Lai and G. G. Coghill. Genetic breeding of control parameters for the hopfield/tank neural net. In *Proceedings IEEE International Joint Conference on Neural Networks*, volume 4, pages 618–632, 1992.

[72] E. L. Lawler, J. K. Lenstra, A. H. Kan, G. Rinnooy, and D. B. Shmoys. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* New York: Wiley, 1985.

[73] O. Lazaro and D. Girma. A Hopfield neural-network-based dynamic channel allocation with handoff channel reservation control. *IEEE Trans. on Vehicular Technology*, 49:1578–1687, 2000.

[74] R. Lee. A transient-chaotic autoassociative network (tcan) based on lee oscillators. *IEEE Trans. on Neural Networks*, 15:1228–1243, 2004.

[75] B. M. Leiner, D. L. Nielson, and F. A. Tobagi. Issues in packet radio network design. In *Proc. IEEE*, volume 75, pages 6–20, 1987.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[76] S. Li and L. P. Wang. Channel assignment for mobile communications using stochastic chaotic simulated annealing. In *Proceedings of the 2001 International Work-Conference on Artificial Neural Networks*, volume 2084, pages 757–776, 2001.

[77] V. Manquinho and J. Marques-Silva. Search pruning techniques in sat-based branch-and-bound algorithms for the binate covering problem. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 21:505–516, 2002.

[78] T. Mizuike and Y. Ito. Optimization of frequency assignment. *IEEE Trans. Communications*, 37:1031–1041, 1989.

[79] H. Nonaka and Y. Kobayashi. Sub-optimal solution screening in optimization by neural networks. In *Proceedings IEEE International Joint Conference on Neural Networks*, volume 4, pages 606–611, 1992.

[80] H. Nozawa. A neural network model as a globally coupled map and applications based on chaos. *Chaos*, 2(3):377–386, 1992.

[81] P. Orponen. Neural networks and complexity theory. In *Mathematical Foundations of Computer Science, MFCS*, pages 50–61, 1992.

[82] L. Page and J. Perry. A practical implementation of the factoring theorem for network reliabilitys. *IEEE Trans. Reliability*, 37(3):259–267, 1988.

[83] I. Parberry. Circuit complexity and neural networks. *MIT Press*, 1994.

[84] C. Peterson and J. R. Anderson. A mean field theory algorithm for neural networks. *Complex Systems*, 1:995–1019, 1987.

[85] C. Peterson and B. Soderberg. *Combinatorial Optimization with Neural Networks*. Blackwell, 1992.

[86] B. Pontano. Interference into angel-modulated systems carrying multichannel telephony signals. *IEEE Trans. Communications*, 21, 1973.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[87] C. Pornavalai, G. Chakraborty, and N. Shiratori. A neural network approach to multicast routing in real-time communication networks. In *Proceedings of International Conference on Network Protocols*, pages 332–339, 1995.

[88] J. Provan. The complexity of reliability computations in planar and acyclic graphs. *SIAM J. Comput.*, 15(3):694–702, 1986.

[89] C. Reeves. *Modern Heuristic Techniques for Combinatorial Problems*. Oxford, Blackwell, 1993.

[90] B. Rosen and R. Nakano. Simulated annealing - basics and recent topics on simulated annealing. *Japan. Soc. Artific. Intellig.*, 9:365–372, 1994.

[91] D. Rumelhart, M. J.L, and P. R. Group. *Parallel Distributed Processing*. MIT Press, Cambridge, MA, 1986.

[92] S. Salcedo-Sanz, C. B. no Calzón, and A. R. Figueiras-Vidal. A mixed neural-genetic algorithm for the broadcast scheduling problem. *IEEE Trans. on Wireless communications*, 2:277–283, 2003.

[93] S. Salcedo-Sanz, R. Santiago-Mozos, and C. B. no Calzón. A hybrid Hopfield network-simulated annealing approach for frequency assignment in satellite communications systems. *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, 34:1108–1116, 2004.

[94] S. Salcedo-Sanz and X. Yao. A hybrid Hopfield network- genetic algorithm approach for the terminal assignment problem. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 34:2343–2354, 2004.

[95] A. Satyanarayana and M. K. Chang. Network reliability and the factoring theorem. *Networks*, 13:107–120, 1983.

[96] K. Sekine and H. Imai. A unified approach via bdd to the network reliability and path numbers. *Tech. Rep. 95-09 Department of Information Science, University of Tokyo*, 1995.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[97] K. Sekine and H. Imai. Counting the number of paths in a graph via bdds. *IEICE Trans. Fundamentals*, E80-A(4):682–688, 1997.

[98] H. Shi and L. P. Wang. A mixed branch-and-bound and neural network approach for the broadcast scheduling problem. In *Design and Application of Hybrid Intelligent Systems, HIS03, the Third International Conference on Hybrid Intelligent Systems*, pages 42–49, Melbourne, Australia, 2003.

[99] H. Shi and L. P. Wang. A noisy chaotic neural network approach for broadcast scheduling problem in packet radio networks. In *The Second International Conference on computational Intelligence, Robotics and Autonomous Systems*, Singapore, 2003.

[100] H. Shi and L. P. Wang. A noisy chaotic neural network approach to topological optimization of a communication network with reliability constraints. In *Proceedings of International Symposium on Neural Networks*, volume 2, pages 230–235, Dalian, China, 2004.

[101] H. Shi and L. P. Wang. A hybrid neural network for optimal tdma transmission scheduling in packet radio networks. In *Proceedings of the International Joint Conference on Neural Networks*, July 31 - August 4, Montreal, Canada, 2005.

[102] H. Shi and L. P. Wang. Optimal tdma scheduling in broadcasting packet radio networks using a gradual noisy chaotic neural network. In *Poceedings of the first International Conference on Neural Computation*, August 27 - 29, Changsha, China, 2005.

[103] J. Sima and P. Orponen. General-purpose computation with neural networks: A survey of complexity theoretic results. *Neural Computation*, 15:2727–2778, 2003.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[104] J. Sima, P. Orponen, and T. Antti-Poika. On the computational complexity of binary and analog symmetric Hopfield nets. *Neural Computation*, 12:2965–2989, 2000.

[105] C. A. Skarda and W. J. Freeman. How brains make chaos in order to make sense of the world. *Brain Behav. Sci.*, 10:161–195, 1987.

[106] K. A. Smith. Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing*, 11(1):15–34, 1999.

[107] P. Somol, P. Pudil, and J. Kittler. Fast branch & bound algorithms for optimal feature selection. *IEEE Trans. on Pattern Analysis and Machine Intelligences*, 26:900–912, 2004.

[108] H. Tang, K. Tan, and Z. Yi. A columnar competitive model for solving combinatorial optimization problems. *IEEE Trans. on Neural Networks*, 15:1568–1573, 2004.

[109] I. Tokuda, K. Aihara, and T. Nagashima. Adaptive annealing for chaotic optimization. *Phys. Rev. E*, 58:5157–5160, 1998.

[110] L. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):401–421, 1979.

[111] A. Varma and Jayadeva. A novel digital neural network for the travelling salesman problem. In *Neural Information Processing, 2002. ICONIP '02*, volume 2, pages 1320–1324, 2002.

[112] G. Wang and N. Ansari. Optimal broadcast scheduling in packet radio networks using mean field annealing. *IEEE Journal on Selected Areas in Communications*, 15:250–260, 1997.

[113] L. Wang, editor. *Minimizing Interference in Cellular Mobile Communications by Optimal Channel Assignment Using Chaotic Simulated Annealing*. Berlin: Springer, 2003.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[114] L. Wang and K. Smith. On chaotic simulated annealing. *IEEE Transactions on Neural Networks*, 9:716–718, 1998.

[115] L. P. Wang. Oscillatory and chaotic dynamics in neural networks under varying operating conditions. *IEEE Transactions on Neural Networks*, 7:1382–1388, 1996.

[116] L. P. Wang, S. Li, F. Y. Tian, and X. J. Fu. A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing. *IEEE Trans. System, Man, Cybern, Part B - Cybernetics*, 34:2119–2125, 2004.

[117] L. P. Wang and J. Ross. Osillations and chaos in neural networks: an exactly solvable model. In *Poceedings of National Academy of Sciences*, volume 87, pages 9467–9471, 1990.

[118] L. P. Wang and F. Tian. Noisy chaotic neural networks for solving combinatorial optimization problems. In *Proc. International Joint Conference on Neural Networks*, volume 4, pages 37–40, 2000.

[119] R. L. Wang, Z. Tang, and Q. Cao. A Hopfield network learning method for bipartite subgraph problem. *IEEE Trans. on Neural Networks*, 15:1458–1465, 2004.

[120] D. Well. Solving degenerate optimization problems using networks of neural oscillators. *Neural Netwoks*, 5:949–959, 1992.

[121] D. J. Whalen. Communications satellites: Making the global village possible. *http://www.hq.nasa.gov/office/pao/History/satcomhistory.html*.

[122] J. Wieselthier, C. M. Barnhart, and A. Ephremides. A neural network approach to routing without interference in multihop radio networks. *IEEE Trans. Communications*, 42(1):166–177, 1994.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[123] G. Wilson and G. Pawley. On the stalibility of the travelling salesman problem algorithm of Hopfield and tank. *Biol. Cybern.*, 58:63–70, 1988.

[124] M. Yamaguti, editor. *Solution of the optimization problem using the neural network model as a globally coupled map*, 1994.

[125] M. Yamguti, editor. *Transient chaotic neural networks and chaotic simulated annealing*. Amsterdam: Elsevier Science Publishers, 1994.

[126] L. Yan and L. P. Wang. Applications of transiently chaotic neural networks to image restoration. In *Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference on*, pages 265–269, 2003.

[127] L. Yan and L. P. Wang. Image restoration using chaotic simulated annealing. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, pages 3060–3064, 2003.

[128] L. Yan, L. P. Wang, and K. Yap. A noisy chaotic neural network approach to image denoising. In *International Conference on Image Processing, ICIP*, pages 1229–1232, 2004.

[129] X. Yao. Finding approximate solutions to np-hard problems by neural networks is hard. *Information Processing Letters*, 41(2):93–98, 1992.

[130] Y. Yao and W. J. Freeman. Model of biological pattern recognition with spatially chaotic dynamics. *Neural Networks*, 3:153–170, 1990.

[131] M. Yeh, J. Lin, and W. Yeh. A new monte carlo method for estimation network reliability. In *Proc. 16th Int'l conf. Computers and Industrial Engineering*, pages 723–726, 1994.

[132] W. Yeh. A new monte carlo method for the network reliability. In *Proceedings of ICITA 2002*, 2002.

[133] J. Yeo, H. Lee, and S. Kim. An efficient broadcast scheduling algorithm for tdma ad-hoc networks. *Computer & Operations Research*, 29:1793–1806, 2002.

ATTENTION: The Singapore Copyright Act applies to the use of this document. Nanyang Technological University Library

[134] S. Zanakis, J. R. Evans, and A. A. Vazacopoulos. Heuristic methods and applications: a categorized survey. *Europ. J. Operat. Res.*, 43:88–110, 1989.

[135] L. Zheng, K. Wang, and K. Tian. An approach to improve wang-smith chaotic simulated annealing. *International Journal of Neural Systems*, 12:363–368, 2002.