

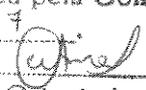
UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E COMPUTAÇÃO
DEPARTAMENTO DE ENG. DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

**UMA ABORDAGEM NEURO-NEBULOSA PARA OTIMIZAÇÃO
DE SISTEMAS E IDENTIFICAÇÃO ROBUSTA**

Autor: Ivan Nunes da Silva

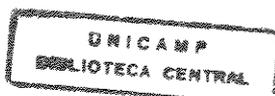
Orientação: Prof. Dr. Wagner Caradori do Amaral

Co-Orientação: Profa. Dra. Lúcia Valéria R. de Arruda

Este exemplar corresponde a redação final da tese defendida por <u>Ivan Nunes da Silva</u> e aprovada pela Comissão Julgada em <u>17 / 12 / 97</u>
 Orientador

Tese apresentada à Faculdade de Engenharia Elétrica e Computação da Universidade Estadual de Campinas, como parte dos requisitos exigidos para obtenção do Título de Doutor em Engenharia Elétrica.

Dezembro de 1997



9805-f000

UNIDADE	BC		
N.º CHAMADA:	T/UNICAMP		
	Si38a		
V.	Ex.		
TOMBO BC/	33009		
PROC.	89.5/98		
C	<input type="checkbox"/>	D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00		
DATA	30/03/98		
N.º CPD			

CM-00106583-1

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Si38a Silva, Ivan Nunes da
 Uma abordagem neuro-nebulosa para otimização de sistemas e identificação robusta / Ivan Nunes da Silva.-- Campinas, SP: [s.n.], 1997.

Orientadores: Wagner Caradori do Amaral, Lúcia Valéria R. de Arruda.

Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes neurais (Computação). 2. Otimização matemática. 3. Identificação de sistemas. 4. Inteligência artificial. I. Amaral, Wagner Caradori do. II. Arruda, Lúcia Valéria R. de. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

AGRADECIMENTOS

Primeiramente agradeço a Deus pela vida e pelos dons que me concedeste. É com muita alegria e satisfação que agradeço aos meus orientadores Prof. Dr. Wagner Caradori do Amaral e Profa. Dra. Lúcia Valéria Ramos de Arruda pela segurança, competência e dedicação demonstradas na orientação deste trabalho, além da amizade cultivada nestes anos de convivência. Agradeço também à Universidade Estadual Paulista (UNESP) por me conceder afastamento parcial de minhas funções docentes, mesmo durante o estágio probatório. Meus sinceros agradecimentos ao CNPq e a FAPESP (Processo 96/06374-2) pelo suporte financeiro concedido a este trabalho desde o início.

Finalmente, gostaria de agradecer aos meus pais pelas lições de vida, caráter e amor demonstrados durante toda a minha vida, e agradecimentos especiais para minha futura esposa Sileide, que sempre me brindou com sua compreensão, carinho e principalmente com seu amor.

Porque *dele* e por meio *dele*, e para *ele*,
são todas as coisas. Glória, pois, a *ele*
eternamente. Amém.

Romanos 11,36.

Para ser grande, sê inteiro.
Nada teu exagera ou exclue.
Sê todo em cada coisa.
Põe quanto és no mínimo que fazes.
Assim em cada lago a lua toda brilha porque
alta vive.

Fernando Pessoa.

Dedico este trabalho para os meus pais,
Gerson e Hosana, pelo incentivo e apoio em
todos os momentos, e para minha futura
esposa *Sileide*, fonte de inspiração de todo
o meu ser.

ABSTRACT

The ability of artificial neural networks to solve complex and diversified problems make them attractive for application in many areas of engineering and science. A neural network is basically composed of many simple processing elements with a high degree of connectivity among them. This thesis presents an architecture of artificial neural network to apply in optimization problems with constraints. More specifically, a topology based in the Hopfield network is proposed to solve some optimization problems, including dynamic programming and combinatorial optimization, and also robust parametric identification problems with unknown-but-bounded disturbance. In this case, the network is used to calculate the parametric uncertainty intervals and the feasible membership set for model parameters.

A rule-based fuzzy system has been developed in order to improve the efficiency and the network convergence to the equilibrium points. The internal parameters of the network, associated with the constraints imposed by a specific problem, are explicitly computed using a valid-subspace technique. Simulation results are provided to illustrate the performance of the proposed approach.

RESUMO

A habilidade de redes neurais artificiais em solucionar problemas complexos e variados, as tornam uma abordagem atrativa a ser aplicada em diversas áreas de engenharia e ciências. Uma rede neural é basicamente composta de inúmeros elementos processadores simples com um alto grau de conectividade entre eles. Esta tese propõe uma arquitetura de uma rede neural artificial aplicável na solução de problemas de otimização com restrições. Mais especificamente, um modelo baseado na rede de Hopfield é proposto para resolver diferentes tipos de problemas de otimização, incluindo problemas de programação dinâmica e otimização combinatorial, e também problemas de identificação robusta com erro desconhecido mas limitado.

Um sistema baseado em lógica nebulosa é desenvolvido com o propósito de conduzir, de forma eficiente, a convergência da rede em direção a seus pontos de equilíbrio. Os parâmetros internos da rede, associados às restrições, são derivados utilizando a técnica de subespaço válido de soluções. Resultados de simulações são apresentados para ilustrar o desempenho da abordagem proposta.

ÍNDICE

Capítulo 1. Introdução

1.1-) Motivação e Relevância do Trabalho.....	01
1.2-) Proposta da Tese	04
1.3-) Organização da Tese	06
Bibliografia.....	08

Capítulo 2. Fundamentos Teóricos e a Rede de Hopfield Modificada

2.1-) Introdução.....	12
2.2-) A Rede de Hopfield Convencional.....	14
2.3-) A Abordagem de Subespaço-Válido	17
2.4-) A Rede de Hopfield Modificada.....	18
2.5-) Dinâmica da Rede de Hopfield Modificada.....	20
2.6-) Descrição do Controlador Lógico Nebuloso	24
2.7-) Metodologia Para Mapeamento de Problemas.....	28
2.8-) Conclusão	29
Bibliografia.....	30

Capítulo 3. Otimização Linear e Quadrática Por Redes Neurais Artificiais

3.1-) Introdução.....	33
3.2-) Problemas de Otimização Linear.....	34
3.2.1-) Mapeamento de Problemas de Programação Linear Através da Rede de Hopfield Modificada.....	34
3.2.2-) Análise de Complexidade das Redes Neurais Artificiais.....	38
3.2.3-) Exemplo de Simulação.....	41
3.2.4-) Considerações Sobre a Aplicação da RHM na Solução de Problemas de Programação Linear.....	45
3.3-) Problemas de Otimização Quadrática.....	45
3.3.1-) Mapeamento de Problemas de Programação Quadrática Através da Rede de Hopfield Modificada.....	45
3.3.2-) Resultados de Simulação.....	47
3.3.3-) Considerações Sobre a Aplicação da RHM na Solução de Problemas de Programação Quadrática.....	49
3.4-) Conclusão	50
Bibliografia.....	50

Capítulo 4. Otimização Não-Linear Através de Redes Neurais Artificiais

4.1-) Introdução.....	52
4.2-) Conceitos Fundamentais de Otimização Não-Linear.....	53
4.3-) Problemas de Otimização Não-Linear.....	55
4.3.1-) Mapeamento de Problemas de Otimização Restritos Através da Rede de Hopfield Modificada.....	55
4.3.2-) Mapeamento de Problemas de Otimização Irrestritos Através da Rede de Hopfield Modificada.....	64
4.4-) Conclusão	67
Bibliografia.....	68

Capítulo 5. Otimização Combinatorial e Prog. Dinâmica Por Redes Neurais Artificiais

5.1-) Introdução.....	70
5.2-) Notações, Definições e Propriedades	71
5.3-) Problemas de Otimização Combinatorial.....	77
5.3.1-) Mapeamento do Problema do Caixeiro Viajante.....	78
5.3.2-) Mapeamento do Problema das N-Rainhas.....	82
5.3.3-) Mapeamento do Problema do Matching-Bipartido.....	89
5.4-) Problemas de Programação Dinâmica.....	94
5.4.1-) Mapeamento de Problemas de Programação Dinâmica Através da Rede de Hopfield Modificada.....	96
5.4.2-) Resultados de Simulação.....	101
5.5-) Conclusão	106
Bibliografia.....	107

Capítulo 6. Identificação Robusta Através de Redes Neurais Artificiais

6.1-) Introdução.....	109
6.2-) Formulação do Problema.....	110
6.3-) Mapeando Problemas de Identificação Robusta Através da Rede de Hopfield Modificada.....	113
6.4-) Simulações de Problemas de Identificação Robusta.....	116
6.4.1-) Identificação Robusta Para Modelos Lineares.....	116
6.4.2-) Identificação Robusta Para Modelos Não-Lineares.....	120
6.5-) Conclusão	123
Bibliografia.....	123

Capítulo 7. Conclusões Gerais e Trabalhos Futuros

7.1-) Resumo da Tese	126
7.2-) Conclusões Gerais.....	127
7.3-) Linhas Futuras de Pesquisas.....	129

Apêndice A	131
-------------------------	-----

Apêndice B	133
-------------------------	-----

Apêndice C	134
-------------------------	-----

Apêndice D	141
-------------------------	-----

Apêndice E	143
-------------------------	-----

LISTA DE FIGURAS

Figura 2.1. A Rede de Hopfield Convencional.....	14
Figura 2.2. A Rede de Hopfield Modificada.....	18
Figura 2.3. Funções de Pertinência do Controlador Nebuloso	27
Figura 3.1. Evolução das Variáveis Para o Problema da Gasolina.....	44
Figura 3.2. Comportamento da Função Objetivo Para o Problema da Gasolina.....	44
Figura 3.3. Evolução das Variáveis Para o Problema de Prog. Quadrática	48
Figura 3.4. Comportamento da Função Objetivo Para o Problema de Prog. Quadrática	48
Figura 4.1. Evolução do Vetor de Saída da Rede (Problema Restrito 1)	60
Figura 4.2. Comportamento da Função Objetivo (Problema Restrito 1).....	60
Figura 4.3. Evolução do Vetor de Saída da Rede (Problema Restrito 2)	61
Figura 4.4. Comportamento da Função Objetivo (Problema Restrito 2).....	62
Figura 4.5. Evolução do Vetor de Saída da Rede (Problema Restrito 3)	63
Figura 4.6. Comportamento da Função Objetivo (Problema Restrito 3).....	63
Figura 4.7. Evolução do Vetor de Saída da Rede (Problema Irrestrito 1).....	65
Figura 4.8. Comportamento da Função Objetivo (Problema Irrestrito 1)	66
Figura 4.9. Evolução do Vetor de Saída da Rede (Problema Irrestrito 2).....	66
Figura 4.10. Comportamento da Função Objetivo (Problema Irrestrito 2)	67
Figura 5.1. Ilustração do Subespaço-Válido.....	76
Figura 5.2. O Problema das N-Rainhas.....	83
Figura 5.3. Soluções do Problema das N-Rainhas (N=5).....	87
Figura 5.4. Evolução da Matriz V Para o Problema das N-Rainhas Com N=5	88
Figura 5.5. Evolução da Matriz V Para o Problema das N-Rainhas Com N=8	89
Figura 5.6. Ilustração de Um Grafo Bipartido	90
Figura 5.7. Grafo Inicial e Grafo Bipartido Mínimo.....	93
Figura 5.8. Evolução da Matriz V Para o Problema do Matching-Bipartido.....	94
Figura 5.9. O Problema de Programação Dinâmica.....	96
Figura 5.10. Mapa de Caminhos Para o Problema Proposto	101
Figura 5.11. Evolução da Matriz V Para o Prob. de Prog. Dinâmica (n=16, m=8).....	104
Figura 5.12. Comportamento Função Obj. Para o Prob. de Prog. Dinâmica (n=16, m=8)...	104
Figura 5.13. Evolução da Matriz V Para o Prob. de Prog. Dinâmica (n=16, m=16).....	105
Figura 5.14. Comportamento Função Obj. Para o Prob. de Prog. Dinâmica (n=16, m=16) .	106
Figura 6.1. A Região de Pertinência Paramétrica.....	111
Figura 6.2. Perspectivas de Seleção do Vetor i^{ot}	115
Figura 6.3. Saída do Modelo em Função de Um Conjunto de Medidas.....	118
Figura 6.4. Região de Pertinência Paramétrica (Modelo Não-Linear)	121
Figura 6.5. Região de Pertinência Paramétrica (1ª Sub-Região).....	122
Figura A.1. Hardware Analógico da Rede de Hopfield.....	132
Figura C.1. A Rede de Kennedy e Chua.....	136

LISTA DE TABELAS

Tabela 2.1. Conjunto de Regras Nebulosas	26
Tabela 3.1. Resultados da Análise de Complexidade dos Algoritmos (PL).....	40
Tabela 3.2. Especificação de Dados do Problema da Gasolina.....	42
Tabela 3.3. Comparação dos Resultados (Problema da Gasolina).....	43
Tabela 3.4. Comparação dos Resultados Para Δt Fixo	43
Tabela 3.5. Comparação de Resultados (Programação Quadrática)	47
Tabela 3.6. Resultados da Análise de Complexidade dos Algoritmos (PQ).....	49
Tabela 5.1. Comparação dos Resultados (Problema do Caixeiro Viajante).....	81
Tabela 5.2. Soluções Para o Problema das N-Rainhas	87
Tabela 5.3. Comparação dos Resultados (Programação Dinâmica).....	102
Tabela 6.1. Intervalos de Incerteza Paramétrica (Processo 1)	117
Tabela 6.2. Estimadores Paramétricos (Processo 1).....	118
Tabela 6.3. Intervalos de Incerteza Paramétrica (Processo 2)	119
Tabela 6.4. Estimadores Paramétricos (Processo 2).....	120
Tabela 6.5. Vetor de Medidas Para o Modelo Não-Linear.....	120
Tabela 6.6. Estimadores Centrais (Modelo Não-Linear).....	122
Tabela D.1. Soluções Para o Problema das N-Rainhas (N=5).....	141
Tabela D.2. Soluções Para o Problema das N-Rainhas (N=8).....	142

CAPÍTULO 1

INTRODUÇÃO

1.1-) MOTIVAÇÃO E RELEVÂNCIA DO TRABALHO

No fim dos anos cinquenta, a cibernética propôs modelar a inteligência humana através de duas filosofias opostas. A primeira foi uma abordagem ‘holística’ que procurava modelar a estratégia de computação do cérebro, utilizando um número elevado de processadores simples. Nesta abordagem, cada processador, envolvido com um estímulo específico, utiliza informações geradas por outros processadores, e fornece uma resposta (baseada nas informações recebidas) que é interpretada por vários outros elementos. A inexistência, na época, de sistemas de computação capazes de processar a teoria holística levou-a ao esquecimento por parte da comunidade científica. A segunda abordagem, ‘reducionista’, utiliza apenas um elemento processador complexo que, atuando sobre informações armazenadas localmente, pode ser usado para dividir cálculos complexos em subproblemas mais simples. Entretanto, a abordagem reducionista não tem sido muito difundida por utilizar mecanismos de processamento que diferem daqueles utilizados pelo cérebro.

Recentemente tem havido um renascimento da abordagem holística sob a forma de ‘modelos conexionistas’, ‘processamento distribuído paralelo’ ou ‘redes neurais artificiais’. Este ressurgimento deve-se ao desenvolvimento de novos algoritmos de aprendizagem e também ao avanço da capacidade de cálculos dos sistemas de computação, que tornam as simulações dos modelos holísticos mais eficientes [Hamm90].

Atualmente, Redes Neurais Artificiais (RNA) representam um domínio de pesquisa multidisciplinar em diversas áreas: física, matemática, engenharia, psicologia, neurofisiologia, biologia, etc. Os princípios fundamentais das redes neurais artificiais são baseados na modelagem biológica de processos neurofisiológicos, cognitivos e comportamentais. Assim, o princípio fundamental para a utilização de RNA na solução de problemas complexos e variados é baseado no funcionamento do cérebro humano, que tem capacidade de resolver de forma rápida e satisfatória, problemas mal definidos e que exigem um grande esforço computacional como, por exemplo reconhecimento de voz e imagens visuais. A principal semelhança entre o sistema neural biológico e uma RNA é a elevada taxa de computação por intermédio de um grande número de processadores (elementos simples de processamento), com elevado grau de conectividade entre eles [Tagl91].

O primeiro modelo matemático do funcionamento de um neurônio foi proposto por McCulloch e Pitts [McCu43] em 1943. O modelo do neurônio de McCulloch é um dispositivo binário, em que a saída é determinada em função da soma ponderada de suas entradas. Este modelo, embora simples, foi inovador e de natureza pioneira, sendo ainda o elemento básico da maioria dos modelos atuais de RNA. No final da década de 1950, Rosenblatt [Rose59] deu prosseguimento as idéias de McCulloch ao introduzir uma rede de neurônios denominada 'Perceptron'. Nesta mesma época, Widrow [Widr60] desenvolveu um modelo neural linear batizado de ADALINE (abreviação de ADaptive LINEar Element) e mais tarde propôs a sua generalização multidimensional, o MADALINE (múltipla ADALINE). A contribuição principal dos trabalhos de Widrow foi a proposta do princípio de treinamento para as redes ADALINES, conhecido como a 'Regra Delta', que foi mais tarde generalizada para redes com modelos neurais mais elaborados [Rume86]. Em 1969, Minsky e Papert [Mins69] publicaram o clássico livro 'Perceptrons - An Introduction to Computational Geometry'. Neste livro, os autores argumentaram enfaticamente quanto às limitações básicas de perceptrons isolados, a começar pela impossibilidade de se implementar regras lógicas tão simples como a do ou-exclusivo. A publicação do livro 'Perceptrons' é assinalada nos anais da história das redes neurais como um divisor de águas que paralisou, a partir de 1970, a pesquisa nesta área.

No início da década de 80, houve uma retomada do interesse em redes neurais artificiais por diversos fatores. Entre os quais, ressaltam-se: melhores conhecimentos da estrutura real do cérebro, disponibilidade de computadores com maior capacidade de cálculo

(inclusive paralelos), e sobretudo o desenvolvimento de novos algoritmos de aprendizado apresentados por Rumelhart, Hinton e Williams [Rume86], Le Cun [LeCu85] e Parker [Park85]. Rumelhart, Hinton e Williams desenvolveram o algoritmo de aprendizado ‘backpropagation’ para as redes do tipo perceptron com estrutura multicamadas. Paralelamente, os fundamentos teóricos foram melhorados, principalmente por Werbos [Werb74], Grossberg [Gros76a, Gros76b], Hopfield [Hopf82] e Kohonen [Koho88], contribuindo também para o ressurgimento da área. Desde então, várias aplicações têm sido mapeadas através de redes neurais artificiais, tais como: reconhecimento de padrões [Widr88], processamento de imagens [Lin91], processamento de sinais [Hayk92], processamento de fala [Waib89], sistemas de controle [Nare90, Ants92], Robótica [Smag91], otimização [Hopf85, Tagl91], identificação de sistemas [Nare90, Silv97], e particionamento de grafos [Cimi96].

Resumindo, as redes neurais artificiais podem ser caracterizadas como modelos computacionais, com propriedades particulares, cujas principais características são definidas como:

- (i) Habilidade de aprender a partir da experiência.
- (ii) Capacidade intrínseca de operação em paralelo.
- (iii) Sistemas inspirados no cérebro humano.
- (iv) Não necessidade de conhecimento a priori de eventuais modelos matemáticos que descrevem determinada aplicação.

Estas características diferem do modelo de computação convencional de von Neumann, em que o processamento da informação é efetuado de forma sequencial e obedecendo um conjunto de procedimentos constituídos por estruturas simbólicas [Hayk94].

Paralelamente ao desenvolvimento das RNA, Zadeh desenvolveu a teoria sobre conjuntos nebulosos na década de 60 [Zade65]. Atualmente, a diversificação de tecnologias advindas da lógica nebulosa tem, como no caso das RNA, permitido sua aplicação em diversas áreas do conhecimento. As técnicas de sistemas nebulosos são especialmente utilizadas nos casos onde não existem modelos matemáticos capazes de descrever precisamente o processo estudado. Estas técnicas fornecem uma estrutura poderosa para manipular informações aproximadas. Assim, o processo pode ser controlado a partir de um conjunto de regras nebulosas do tipo ‘Se-Então’, capaz de tratar (coordenar) conhecimentos incompletos, incertos

ou mesmo conflitantes [Lee90]. Entre as aplicações dos sistemas nebulosos, pode-se destacar: controle de processos industriais e aplicações em produtos de consumo (ar condicionado, câmeras de vídeo, fornos de microondas, etc.) [Gomi94], em sistemas industriais (controle de grupo de elevadores, ventilação de túneis urbanos, controle de tráfego urbano, etc.) [Hiro94], em medicina [Adla86], e em sistemas especialistas [Gain85]. Outros exemplos de aplicações podem ser encontrados em [Hiro93, Klir88, Maie85].

Em resumo, as redes neurais e os sistemas baseados em lógica nebulosa são ferramentas que permitem o desenvolvimento de algoritmos para modelar e controlar sistemas dinâmicos complexos. No entanto, uma desvantagem inerente à arquitetura das redes neurais é a dificuldade em se conhecer especificamente como esta rede resolve uma determinada tarefa de controle. Geralmente, não é possível recuperar qualquer tipo de conhecimento estrutural que possa ser formulado em termos de regras, ou através de alguma outra forma de representação própria da inteligência artificial [Nauc93]. Também é difícil a inclusão na estrutura da rede de algum conhecimento a priori que possa reduzir o seu tempo de convergência ou aprendizado. Por outro lado, o uso de um controlador nebuloso permite interpretar o comportamento do sistema de controle através de regras linguísticas explícitas que compõem o controlador. Assim, os sistemas neuro-nebulosos têm se desenvolvido como uma abordagem promissora que incorpora, num sistema integrado, as vantagens das redes neurais e dos sistemas baseados em lógica nebulosa.

Dentro deste contexto, este trabalho é voltado ao desenvolvimento de uma estrutura de redes neurais que utiliza um sistema baseado em lógica nebulosa com o propósito de diminuir o seu tempo de convergência, melhorando assim a sua eficiência, e de incorporar informações a priori sobre os parâmetros de controle e ponderação da rede.

1.2-) PROPOSTA DA TESE

O objetivo desta tese está relacionado com a aplicação de redes neurais artificiais e de técnicas de sistemas nebulosos na solução de problemas de otimização e de identificação de sistemas. Utiliza-se uma rede neural com topologia recorrente para mapear e solucionar problemas de otimização, de naturezas diferentes, e problemas de identificação robusta. Dentro

deste contexto, uma rede de Hopfield modificada é desenvolvida nesta tese, com o propósito de resolver sete tipos de problemas citados a seguir:

- Problemas de Otimização Linear.
- Problemas de Otimização Quadrática.
- Problemas de Otimização Combinatorial.
- Problemas de Programação Dinâmica.
- Problemas de Otimização Não-Linear (restritos e irrestritos).
- Problemas de Identificação Robusta Para Modelos Lineares.
- Problemas de Identificação Robusta Para Modelos Não-Lineares.

Atualmente, utilizam-se arquiteturas de redes diferentes, ou sistemáticas (metodologias) de mapeamento distintas, com o intuito de mapear cada um dos vários tipos de problemas citados acima [Chiu91, Hopf85, Kenn88, Maa92, Rome93, Zhan92]. Algumas destas arquiteturas apresentam dificuldades no mapeamento de problemas com restrição, resultando em soluções, algumas vezes, ineficazes.

Ao contrário, a utilização da rede de Hopfield modificada proposta nesta tese, permite o mapeamento de problemas de otimização e identificação, de naturezas diferentes, utilizando uma mesma metodologia, independente do problema tratado. No caso dos problemas com restrição, estas são facilmente mapeadas na rede proposta via técnica do subespaço-válido de soluções [Aiyee90]. Os benefícios e justificativas em utilizar a arquitetura proposta são enumerados a seguir:

- (i) Portabilidade da arquitetura em relação à natureza dos problemas.
- (ii) Maior simplicidade computacional para mapear os problemas.
- (iii) Facilidade de implementação em hardware.
- (iv) Viabilização do tratamento das restrições envolvidas com os problemas.
- (v) Plasticidade em relação às condições requeridas para o mapeamento dos problemas.
- (vi) Não necessidade de definição de parâmetros, de controle ou ponderação, para sua inicialização.

Esta lista de benefícios comprova que a sintetização de uma estrutura neuro-nebulosa capaz de solucionar diversos tipos de problemas é uma ferramenta poderosa e vantajosa, além de fornecer um novo método para solução destes problemas. Sob este aspecto, o cálculo dos

parâmetros internos da rede de Hopfield modificada, associados com o mapeamento de restrições, são explicitamente computados utilizando a técnica denominada subespaço-válido de soluções. Esta técnica fornece um modo prático para cálculo dos parâmetros internos da rede em relação a cada problema a ser mapeado por esta.

1.3-) ORGANIZAÇÃO DA TESE

Esta tese está dividida em 7 capítulos. A organização dos capítulos é feita como segue:

No Capítulo 2, apresenta-se os conceitos fundamentais relativos às redes neurais artificiais com enfoque principal sobre a rede de Hopfield modificada. Em seguida, são descritos, a técnica de subespaço-válido de soluções e os fundamentos principais de sistemas nebulosos.

No Capítulo 3, desenvolve-se o mapeamento de problemas de otimização linear e quadrática através da rede de Hopfield modificada. Uma análise da complexidade do algoritmo neural é realizada e comparada a outras topologias de redes.

No Capítulo 4, descreve-se como é feito o mapeamento de problemas de otimização não-linear restritos e irrestritos.

No Capítulo 5, apresenta-se o mapeamento de problemas de otimização combinatorial e de problemas de programação dinâmica utilizando a rede de Hopfield modificada. É também realizada, uma análise comparativa com outros métodos existentes na literatura.

No Capítulo 6, faz-se o mapeamento de problemas de estimação paramétrica robusta para modelos lineares e não-lineares.

No Capítulo 7, apresentam-se as conclusões gerais relativas a esta tese, e traça-se as linhas futuras de pesquisas.

A maioria dos tópicos relacionados nesta tese já foram divulgados, ou estão sendo julgados para publicação. Algumas das publicações são relacionadas a seguir:

Parte do Capítulo 2 foi submetida ao:

- Ivan N. da Silva, Wagner C. do Amaral, Lúcia V. R. de Arruda, “A novel approach to robust parameter estimation using neurofuzzy systems”. Submitted to: *Mathematics and Computers in Simulation*.

O conteúdo do Capítulo 3 foi publicado em:

- Ivan N. da Silva, Lúcia V. R. de Arruda, Wagner C. do Amaral. “Uma rede neural para resolver problemas de programação linear com variáveis limitadas”. *Proc. of the 1996 Latin-Iberian-American Congress on Operations Research and Systems Engineering*, Rio de Janeiro-RJ, pp. 625-630, Setembro de 1996.

O conteúdo do Capítulo 4 foi submetido ao:

- Ivan N. da Silva, Wagner C. do Amaral, Lúcia V. R. de Arruda, “Nonlinear optimization by artificial neural networks”. Submitted to: *1998 IEEE World Congress on Computational Intelligence*, Alaska, May 1998.

O conteúdo do Capítulo 5 foi registrado nos seguintes trabalhos:

- Ivan N. da Silva, Wagner C. do Amaral, Lúcia V. R. de Arruda. “A new method for mapping dynamic programming problems using neural networks”. *Proc. of the 1997 World Multiconference on Systemics, Cybernetics and Informatics (Invited Session)*, Caracas, Venezuela, pp. 311-315, July 1997.
- Ivan N. da Silva, Wagner C. do Amaral, Lúcia V. R. de Arruda. “An artificial neural network approach for solving dynamic programming problems”. Submitted to: *IEEE Transactions on Neural Networks*.
- Ivan N. da Silva, Wagner C. do Amaral, Lúcia V. R. de Arruda. “Resolvendo problemas de otimização combinatorial utilizando redes neurais artificiais - O Problema das N-Rainhas”. *Anais do III Congresso Brasileiro de Redes Neurais*, pp. 443-448, Florianópolis, SC, Julho de 1997.

O conteúdo do Capítulo 6 foi publicado em:

- Ivan N. da Silva, Wagner C. do Amaral, Lúcia V. R. de Arruda, “Robust estimation of parametric membership regions using artificial neural networks”. *International Journal of Systems Sciences*, vol. 28, no. 5, pp. 447-455, 1997.
- Ivan N. da Silva, Lúcia V. R. de Arruda, Wagner C. do Amaral. “Aplicação da rede de Hopfield na estimação de regiões de pertinência paramétrica para modelos lineares”. *Revista Controle e Automação*, vol. 7, no. 3, pp. 118-125, 1996.

- Ivan N. da Silva, Wagner C. do Amaral, Lúcia V. R. de Arruda. “Robust parametric estimation for nonlinear models using artificial neural networks”. *Proc. of the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, Florida, October 1997.
- Ivan N. da Silva, Lúcia V. R. de Arruda, Wagner C. do Amaral. “A neurofuzzy method to robust parametric estimation with unknown-but-bounded error”. *Proc. of the 1996 IEEE International Conference on Systems, Man, and Cybernetics*, Beijing, China, pp. 351-355, October 1996.
- Ivan N. da Silva, Lúcia V. R. de Arruda, Wagner C. do Amaral. “Application of the Hopfield network in robust estimation of parametric membership sets for linear models”. *Proc. of the 1995 IEEE International Conference on Systems, Man, and Cybernetics*, Vancouver, Canada, pp. 3949-3953, October 1995.

BIBLIOGRAFIA

- [Adla86] K. -P. Adlassnig. “Fuzzy set theory in medical diagnosis”. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-16, pp. 260-265, 1986.
- [Aiye90] S. V. B. Aiyer, M. Niranja and F. Fallside. “A theoretical investigation into the performance of the Hopfield model”. *IEEE Transaction on Neural Networks*, vol. 1, no. 2, pp. 204-215, 1990.
- [Ants92] P. J. Antsaklis. “Neural networks in control systems”. *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 8-10, 1992.
- [Cimi96] R. Cimikowski and P. Shope. “A neural-network algorithm for a graph layout problem”. *IEEE Trans. on Neural Networks*, vol. 7, no. 3, pp. 341-345, 1996.
- [Chiu91] C. Chiu, C. Y. Maa and M. A. Shanblatt. “Energy function analysis of dynamic programming neural networks”. *IEEE Trans. on Neural Networks*, vol. 2, no. 4, pp. 418-426, 1991.
- [Gain85] B. R. Gaines and M. L. G. Shaw. “From fuzzy logic to expert systems”. *Information Sciences*, vol. 36, pp. 5-16, 1985.

- [Gomi94] F. A. C. Gomide, R. R. Gudwin. “Modelagem, controle, sistemas e lógica nebulosa”. *Revista Controle e Automação*, vol. 4, no. 3, pp. 97-115, 1994.
- [Gros76a] S. Grossberg. “Adaptive pattern classification and universal recording: I. Parallel development and coding of neural detectors”. *Biological Cybernetics*, vol. 23, pp. 121-134, 1976.
- [Gros76b] S. Grossberg. “Adaptive pattern classification and universal recording: II. Feedback, expectation, olfaction, illusions”. *Biological Cybernetics*, vol. 23, pp. 187-202, 1976.
- [Hamm90] D. Hammerstrom. “A VLSI architecture for high-performance, low-cost, on-chip learning”. *Proc. of the Int. Conf. on Neural Networks*, San Diego, CA, pp. 537-544, 1990.
- [Hayk92] S. Haykin and H. Leung. “Blind equalization formulated as a self-organized learning process”. *Proc. of the Twenty-Sixth Asilomar Conference on Signals, Systems, and Computers*, pp. 346-350, Pacific Grove, CA, 1992.
- [Hayk94] S. Haykin. *Neural Networks - A Comprehensive Foundation*. McMillan Inc., Englewood Cliffs, NJ, 1994.
- [Hiro93] K. Hirota. *Industrial Applications of Fuzzy Technology*. Springer-Verlag, New York, NY, 1993.
- [Hiro94] K. Hirota. “Industrial applications and education on fuzzy systems in Japan”. *Revista Controle e Automação*, vol. 4, no. 3, pp. 155-161, 1994.
- [Hopf82] J. J. Hopfield. “Neural network and physical systems with emergent collective computational abilities”. *Proc. of the National Academy of Sciences of the U.S.A.*, vol. 79, pp. 2554-2558, 1982.
- [Hopf85] J. J. Hopfield and D. W. Tank. “Neural computation of decisions in optimization problems”. *Biological Cybernetics*, vol. 52, pp. 141-152, 1985.
- [Kenn88] M. P. Kennedy and L. O. Chua. “Neural network for nonlinear programming”. *IEEE Trans. on Circuits and Systems*, vol. 35, no. 5, pp. 554-562, 1988.
- [Klir88] G. J. Klir and T. A. Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall, Englewood Cliffs, NJ, 1988.

- [Koho88] T. Kohonen. "An introduction to neural network". *Neural Networks*, vol. 1, no. 1, pp. 3-16, 1988.
- [LeCu85] Y. Le Cun. "Une procedure d'apprentissage pour reseau a seuil assymétrique". *Cognitiva*, vol. 85, pp. 599-604, 1985.
- [Lee90] C. C. Lee. "Fuzzy logic in control systems: fuzzy logic controller, Part I and II". *IEEE Trans. on Systems, Man and Cybernetics*, vol. 20, pp. 404-435, 1990.
- [Lin91] W. C. Lin, F. Y. Liao, C. K. Tsao and T. Lingutla. "A hierarchical multiple view approach to three-dimensional object recognition". *IEEE Trans. on Neural Networks*, vol. 2, pp. 84-92, 1991.
- [Maa92] C. Y. Maa and M. A. Shamblat. "Linear and quadratic programming neural network analysis". *IEEE Trans. on Neural Networks*, vol. 33, no. 4, 1992.
- [Maie85] J. Maiers and Y. S. Sherif. "Applications of fuzzy set theory". *IEEE Trans. on Systems, Man and Cybernetics*, vol. 15, pp. 175-186, 1985.
- [McCu43] W. S. McCulloch and W. Pitts. "A logical calculus of the ideas immanent in nervous activity". *Bulletin of Mathematical Biophysics*, vol. 5, 115-133, 1943.
- [Mins69] M. L. Minsky and S. A. Papert. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, Cambridge, Massachusetts, 1969.
- [Nauc93] D. Nauck and R. Kruse. "A fuzzy neural network learning fuzzy control rules and membership functions by fuzzy error backpropagation". *Proc. of the IEEE International Conference on Neural Networks*, San Francisco, CA, pp. 1022-1027, 1993.
- [Nare90] K. S. Narendra and K. Parthasarathy. "Identification and control of dynamical systems using neural networks". *IEEE Trans. on Neural Networks*, vol. 1, pp. 4-27, 1990.
- [Park85] D. B. Parker. *Learning Logic: Casting the Cortex of the Human Brain in Silicon*. Technical Report TR-47, Center of Computational Research in Economics and Management Science, Cambridge, Massachusetts, 1985.
- [Rome93] R. A. F. Romero. *Otimização de Sistemas Através de Redes Neurais Artificiais*. Tese de Doutorado, DCA/FEE/UNICAMP, Campinas-SP, Julho de 1993.

- [Rose59] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, DC, 1959.
- [Rume86] D. E. Rumelhart, G. E. Hinton and R. J. Williams. “Learning internal representations by error propagation”. In: *Parallel Distributed Processing*, vol. 1, chap. 8, The MIT Press, Cambridge, Massachusetts, 1986.
- [Silv97] I. N. Silva, W. C. Amaral and L.V.R. Arruda. “Robust estimation of parametric membership regions using artificial neural networks”. *International Journal of Systems Science*, vol. 28, no. 5, pp. 447-455, 1997.
- [Smag91] P. P. van der Smagt and B. J. A. Krose. “A real-time learning neural robot controller”. *Proc. of the 1991 International Conference on Artificial Neural Network*, Espoo, Finland, pp. 351-356, 1991.
- [Tagl91] G. A. Tagliarini, J. F. Christ and E. W. Page. “Optimization using neural networks”. *IEEE Trans. on Computers*, vol. 40, no. 12, pp. 1347-1358, 1991.
- [Waib89] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and Lang K. “Phoneme recognition using time delay neural networks”. *IEEE Trans. on Acoustics, Speech and Signal Processing*, ASSP-37, pp. 328-339, 1989.
- [Werb74] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph.D. Thesis, Harvard University, Cambridge, Massachusetts, 1974.
- [Widr60] B. Widrow and M. E. Hoff. “Adaptive switching circuits”. *IRE WESCON Convention Record*, Part 4, pp. 96-104, 1960.
- [Widr88] B. Widrow, R. G. Winter and R. A. Baxter. “Layered neural nets for pattern recognition”. *IEEE Trans. on Acoustics, Speech, and Signal Processing*. vol. 36, pp. 1109-1117, 1988.
- [Zade65] L. A. Zadeh. “Fuzzy sets”. *Information and Control*, vol. 8, pp. 338-353, 1965.
- [Zhan92] S. Zhang, X. Zhu and L. -H. Zou. “Second-order neural nets for constrained optimization”. *IEEE Trans. on Neural Networks*, vol. 3, no. 6, pp. 1021-1024, 1992.

CAPÍTULO 2

FUNDAMENTOS TEÓRICOS E A REDE DE HOPFIELD MODIFICADA

2.1-) INTRODUÇÃO

Sistemas baseados em Redes Neurais Artificiais possuem elevadas taxas de computação por utilizarem um número massivo de elementos processadores simples com alta taxa de conectividade entre si. Como as Redes Neurais Artificiais (RNA) também procuram representar a arquitetura do cérebro humano em hardware eletrônico, os sistemas baseados em RNA oferecem um novo método para solucionar problemas variados e complexos, com um potencial significativo para implementação em hardware (Apêndice A).

Este trabalho está intrinsecamente relacionado com a aplicação de RNA na área de otimização e identificação de sistemas. Mais especificamente, uma rede de Hopfield modificada é desenvolvida com a finalidade de resolver problemas relacionados a estas áreas. A principal vantagem da topologia desenvolvida é permitir que problemas de naturezas diferentes possam ser resolvidos com a utilização de uma mesma estrutura de rede. A análise desta estrutura também permite avaliar a dinâmica envolvida na convergência da rede. A partir desta análise, é possível determinar o comportamento da rede quando esta tende em direção aos pontos de equilíbrio que representam as soluções do problema considerado. O ganho advindo da simplicidade relativa aos aspectos computacionais e da portabilidade da estrutura proposta justificam a utilização da arquitetura desenvolvida neste trabalho.

A estrutura das redes neurais foi desenvolvida a partir de modelos conhecidos de sistemas nervosos biológicos e do próprio cérebro. Os elementos computacionais ou unidades processadoras, denominadas *neurônios artificiais*, correspondem aos nós da rede e são modelos simplificados dos *neurônios biológicos*. Estes modelos foram obtidos a partir da análise da geração e propagação de impulsos elétricos pela membrana celular dos neurônios [Hodg52]. Os neurônios utilizados nos modelos de redes neurais artificiais são não lineares, tipicamente analógicos, e realizam funções simples, como coletar os sinais existentes em suas entradas, agregá-los de acordo com sua função de entrada e produzir uma saída de acordo com sua função de ativação (saída) inerente. O modelo de neurônio mais simples e que engloba as principais características de uma rede neural biológica, paralelismo e alta conectividade, foi proposto por McCulloch e Pitts [McCu43]. Este modelo realiza a soma algébrica ponderada das entradas de um neurônio, que em seguida serve como entrada para a função de ativação não linear, determinando a saída da rede [Hayk94].

Uma rede neural é definida pelo tipo (ou função) do neurônio presente, por sua topologia (ou arquitetura) e pelo algoritmo de aprendizagem utilizado. Este algoritmo deve especificar as condições iniciais da rede e a metodologia de ajuste de seus parâmetros internos para se obter o desempenho desejado. Do ponto de vista funcional, uma rede pode ser Homogênea, quando todos os neurônios têm o mesmo comportamento, ou Heterogênea, caso contrário. Do ponto de vista da topologia da interligação entre neurônios, uma rede pode ser de Alimentação para Frente (Feed-Forward) ou Recorrente (Dinâmica). Nas redes de alimentação para frente, os neurônios são organizados em camadas, e a informação desloca-se em um único sentido, entre camadas adjacentes. Nas redes recorrentes, não existe direção privilegiada para a propagação da informação, podendo haver retroalimentação. Um estudo abrangente sobre os vários tipos de redes neurais artificiais é apresentado em [Hayk94].

A rede desenvolvida nesta tese é uma rede recorrente do tipo Hopfield que utiliza a técnica denominada subespaço-válido de solução para cálculo de seus parâmetros, e contém um sistema lógico nebuloso para ajustar a taxa de convergência da rede. Por ser uma rede recorrente e, portanto, um sistema dinâmico, sua estrutura pode ser representada por um conjunto de equações diferenciais, possibilitando o uso das técnicas clássicas de análise de sistemas dinâmicos no estudo da convergência e estabilidade desta rede.

Na Seção 2.2, descreve-se os aspectos fundamentais relativos à rede de Hopfield convencional. A Seção 2.3 introduz o conceito de subespaço-válido de soluções utilizado para calcular os parâmetros internos da rede. A rede de Hopfield modificada é definida na Seção 2.4, e na Seção 2.5 analisa-se o comportamento dinâmico desta rede. Na Seção 2.6, apresenta-se o controlador lógico nebuloso, desenvolvido para acelerar a taxa de convergência da rede. A Seção 2.7 descreve a metodologia utilizada para o mapeamento de problemas através da rede de Hopfield modificada.

2.2-) A REDE DE HOPFIELD CONVENCIONAL

A rede de Hopfield é provavelmente o melhor exemplo conhecido de rede recorrente [Hush93]. Como definida em [Hopf84, Hopf85], esta rede apresenta normalmente uma única camada com conexões realimentadas entre os nós (Figura 2.1). Na maioria dos casos, os nós (neurônios) são completamente interconectados, ou seja, todos os neurônios da rede são conectados a todos os outros e a si próprio. A equação nodal para a rede de Hopfield contínua no tempo com N-neurônios é dada por:

$$\dot{u}_i(t) = -\eta \cdot u_i(t) + \sum_{j=1}^N T_{ij}(t) \cdot v_j(t) + i_i^b \quad (2.1)$$

$$v_i(t) = g_i(u_i(t)) \quad (2.2)$$

onde: $u_i(t)$ é o estado corrente do i -ésimo neurônio.

$v_i(t)$ é a saída do i -ésimo neurônio.

T_{ij} é o peso conectando o i -ésimo neurônio ao j -ésimo neurônio.

i_i^b é a entrada do i -ésimo neurônio (input bias).

$\eta \cdot u_i(t)$ é um termo de decaimento passivo.

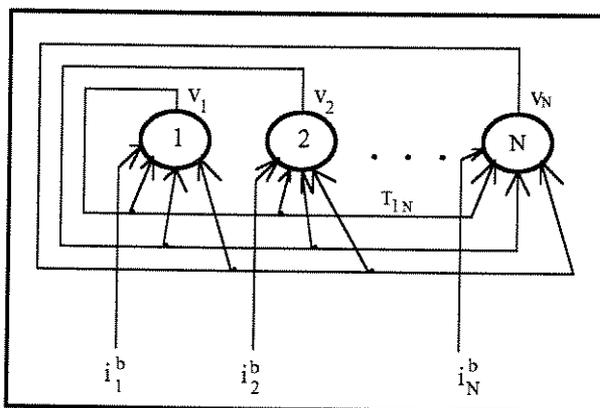


Figura 2.1. A Rede de Hopfield Convencional

Na equação (2.2), $g_i(u_i(t))$ é uma função de ativação, monotonicamente crescente, que limita a saída de cada neurônio em um intervalo pré-definido. Os três tipos básicos de função de ativação, geralmente empregados nas RNA, são [Hayk94]: função ‘threshold’ (sinal ou limiar), função ‘piecewise-linear’ (rampa) e função ‘sigmoid’ (logística ou tangente hiperbólica). Uma versão discreta da rede de Hopfield é obtida quando $\eta=0$ e $T_{ij} = T_{ji}$ [Hopf84, Silv95]. Neste caso, os pontos de equilíbrio da rede correspondem aos valores de $v(t)$ que minimizam uma função de energia da rede (função de Lyapunov). Se os neurônios forem alterados de forma assíncrona, a função de energia associada à rede de Hopfield é definida por [Silv95]:

$$E(t) = -\frac{1}{2} v(t)^T \cdot T \cdot v(t) - v(t)^T \cdot i^b \quad (2.3)$$

O mapeamento de problemas de otimização utilizando uma rede de Hopfield consiste em determinar, independente da natureza diferente de cada problema, a matriz de pesos T (simétrica) e o vetor de entradas i^b associados à função de energia da rede (2.3). Esta função por sua vez deve ser associada à função objetivo e às restrições descrevendo o problema a ser resolvido.

A dificuldade em mapear problemas de otimização variados através de uma rede de Hopfield deve-se à necessidade de satisfazer as várias restrições que são impostas por cada tipo de problema como um todo. Neste caso, a função de energia associada à rede deve levar em conta não só a função objetivo do problema de otimização bem como todas as restrições existentes. A rede atua com o propósito de minimizar simultaneamente a função de energia E^{ot} correspondente à função objetivo e o conjunto de funções (E_k^{rest}) descrevendo as k -ésimas restrições envolvidas no problema. Se qualquer uma destas restrições for violada, a solução é considerada inválida [Gee91]. Uma técnica simples de mapeamento consiste em incluir as restrições como termos na função de energia que são minimizados quando as restrições são satisfeitas, ou seja:

$$E(t) = E^{ot}(t) + c_1 \cdot E_1^{rest}(t) + c_2 \cdot E_2^{rest} + \dots + c_k \cdot E_k^{rest} \quad (2.4)$$

onde c_i são constantes positivas que ponderam cada uma das restrições.

Os primeiros resultados publicados relativos à utilização de redes neurais artificiais, em problemas de otimização, foram fornecidos por Hopfield e Tank em [Hopf85]. Neste trabalho

são descritos os resultados obtidos, em simulações por computador, da rede de Hopfield para resolver o clássico problema do caixeiro viajante. Entretanto, a maioria das soluções obtidas pela rede eram consideradas inválidas. As principais razões para este problema de convergência, como discutido posteriormente por alguns autores, são:

- Dificuldade de se obter os valores corretos para as constantes arbitrárias (c_1, c_2, \dots, c_k) que ponderam os termos de energia relativos às restrições do problema a ser resolvido [Hegd88, Kamg90].
- Influência dos termos de restrições no termo de otimização, dificultando a convergência da rede [Pete89, Wils88].

Em resumo, no processo de otimização com uma rede de Hopfield, cuja função objetivo é dada por (2.4), a multiplicidade de termos de restrições nesta equação afeta consideravelmente o valor da solução final. Como resultado, as soluções obtidas no final do processo de otimização podem ser ineficazes; além disso, o desempenho da rede é sensível aos valores dos parâmetros c_i em (2.4).

Para contornar estes problemas, realiza-se em [Aiy90] uma análise dos autovetores da rede durante a sua convergência, o que permite verificar como E^{ot} e os termos de restrições E_k^{rest} em (2.4) podem ser efetivamente separados dentro de subespaços diferentes, de modo a evitar a infactibilidade do problema. O subespaço que agrupa todas as restrições impostas pelo problema é denominado subespaço-válido de soluções (Seção 2.3). Neste caso, a equação definida por (2.4) é simplificada para:

$$E(t) = E^{ot}(t) + c_0 \cdot E^{conf} \quad (2.5)$$

onde o termo E^{conf} satisfaz todas as restrições E_k^{rest} dadas na equação (2.4). Similarmente a (2.3), as funções de energia E^{ot} e E^{conf} dadas na equação (2.5) passam a ser definidas por:

$$E^{ot}(t) = -\frac{1}{2} \mathbf{v}(t)^T \cdot \mathbf{T}^{ot} \cdot \mathbf{v}(t) - \mathbf{v}(t)^T \cdot \mathbf{i}^{ot} \quad (2.6)$$

$$E^{conf}(t) = -\frac{1}{2} \mathbf{v}(t)^T \cdot \mathbf{T}^{conf} \cdot \mathbf{v}(t) - \mathbf{v}(t)^T \cdot \mathbf{i}^{conf} \quad (2.7)$$

Devido à necessidade de assegurar que E^{conf} seja dominante (satisfação das restrições) sobre E^{ot} deve-se atribuir um valor elevado para a constante c_0 em (2.5). Esta condição torna

a simulação da rede ineficiente, visto que a maioria do tempo será despendido no confinamento de \mathbf{v} dentro do subespaço-válido. Para evitar este problema, propõe-se na Seção 2.4 uma rede de Hopfield modificada que diretamente garante a validade das restrições agrupadas por E^{conf} , dispensando a utilização da constante de ponderação c_0 .

2.3-) A ABORDAGEM DE SUBESPAÇO-VÁLIDO

Para uma grande variedade de problemas com restrições que podem ser resolvidos pela rede de Hopfield, através da decomposição da função de energia da rede (2.3) nas duas parcelas dadas em (2.6) e (2.7), verifica-se que os pontos de equilíbrio da rede, correspondentes aos valores de \mathbf{v} que minimizam a função de energia E^{conf} dada pela equação (2.7), pertencem todos a um mesmo subespaço-comum [Aiy90]. Este subespaço comum, denominado subespaço-válido de soluções, possui equação definida por:

$$\mathbf{v} = \mathbf{T}^{\text{val}} \cdot \mathbf{v} + \mathbf{s} \quad (2.8)$$

onde:

- \mathbf{T}^{val} é uma matriz projeção (isto é: $\mathbf{T}^{\text{val}} \cdot \mathbf{T}^{\text{val}} = \mathbf{T}^{\text{val}}$) que projeta o vetor \mathbf{v} dentro do subespaço-válido ($\mathbf{v}^{\text{val}} = \mathbf{T}^{\text{val}} \cdot \mathbf{v}$, onde \mathbf{v}^{val} é a componente de \mathbf{v} projetada sobre o subespaço-válido).
- \mathbf{s} é um vetor que está relacionado com as restrições do problema a ser resolvido. Este vetor é ortogonal ao subespaço-válido ($\mathbf{T}^{\text{val}} \cdot \mathbf{s} = \mathbf{0}$).

A partir da abordagem do subespaço-válido, a operação da rede de Hopfield é realizada em três passos principais [Silv95]:

- (i) A rede é inicializada em um estado aleatório tal que \mathbf{v} seja limitado pela função de ativação dos neurônios.
- (ii) A minimização de E^{conf} , por intermédio da projeção de \mathbf{v} no subespaço-válido, move \mathbf{v} em direção a um dos pontos de equilíbrio da rede que satisfaz todas as restrições impostas pelo problema (Seção 2.5).
- (iii) A minimização de E^{ot} , garante que o ponto de equilíbrio encontrado no passo (ii) corresponde à solução de ‘custo ótimo’ para o problema.

Portanto, a função principal da projeção no subespaço-válido é confinar todas as restrições impostas pelo problema dentro de um único subespaço. As soluções que são factíveis a todas as restrições estão contidas no subespaço-válido. Um estudo mais abrangente, tratando da abordagem de subespaço-válido, é apresentado em [Silv95].

2.4-) A REDE DE HOPFIELD MODIFICADA

Como descrito anteriormente, a rede de Hopfield convencional tem um tempo de processamento elevado, porque necessita de um esforço computacional adicional para forçar o confinamento de v no subespaço-válido.

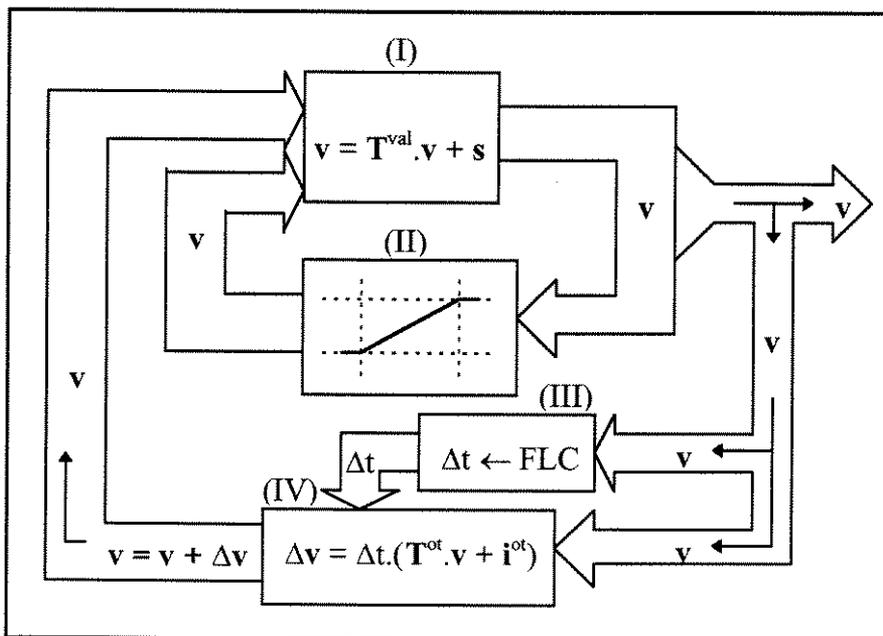


Figura 2.2. A Rede de Hopfield Modificada

Com o propósito de contornar estes problemas propõe-se a implementação da rede de Hopfield modificada com taxa de convergência (passo de iteração) calculada por um controlador lógico nebuloso.

A dinâmica desta Rede de Hopfield Modificada (RHM) pode ser explicitada através de quatro passos (Figura 2.2):

(I) Minimização de E^{conf} : corresponde à projeção de v sobre o subespaço-válido gerado por todas as restrições impostas pelo problema:

$$v = T^{val}.v + s \tag{2.9}$$

Dado que T^{val} é uma matriz projeção, qualquer componente de \dot{v} que seja ortogonal a $T^{val}.v$ é desprezada. Esta operação realiza uma minimização indireta de E^{conf} .

(II) Aplicação de uma função de ativação do tipo ‘rampa-simétrica’, restringindo \mathbf{v} dentro de um hipercubo inicial:

$$g_i^r(v_i) = \begin{cases} \lim_i^{\text{inf}} & \text{se } \lim_i^{\text{inf}} > v_i \\ v_i & \text{se } \lim_i^{\text{inf}} \leq v_i \leq \lim_i^{\text{sup}} \\ \lim_i^{\text{sup}} & \text{se } v_i > \lim_i^{\text{sup}} \end{cases} \quad (2.10)$$

onde $v_i \in [\lim_i^{\text{inf}}, \lim_i^{\text{sup}}]$.

(III) Determinação da taxa de convergência Δt (tamanho do passo de iteração) da rede através do controlador lógico nebuloso (O controlador lógico nebuloso é desenvolvido na Seção 2.6).

(IV) Minimização de E^{ot} : consiste na alteração de \mathbf{v} em direção a uma solução de custo ótimo (definida por \mathbf{T}^{ot} e \mathbf{i}^{ot}) correspondendo aos pontos de equilíbrio da rede, que são as soluções para o problema de otimização considerado. Utilizando a função de ativação rampa-simétrica e dado que $\eta=0$, a equação (2.2) torna-se [Li89]:

$$\mathbf{v}(t) = g_i^r(\mathbf{u}(t)) = \mathbf{u}(t)$$

e comparando (2.1) e (2.6) tem-se:

$$\begin{aligned} \frac{d\mathbf{v}(t)}{dt} = \dot{\mathbf{v}} &= -\frac{\partial E^{\text{ot}}(t)}{\partial \mathbf{v}} \\ \Delta \mathbf{v} &= -\Delta t \cdot \nabla E^{\text{ot}}(\mathbf{v}) = \Delta t \cdot (\mathbf{T}^{\text{ot}} \cdot \mathbf{v} + \mathbf{i}^{\text{ot}}) \end{aligned} \quad (2.11)$$

A minimização de E^{ot} consiste na alteração de \mathbf{v} na direção oposta ao gradiente da função E^{ot} em relação a \mathbf{v} . Estes resultados são também válidos quando uma função de ativação do tipo tangente hiperbólica for utilizada [Abe93].

As principais diferenças existentes entre a rede de Hopfield modificada, desenvolvida nesta seção, em relação a rede de Hopfield convencional são:

- A taxa de convergência da rede de Hopfield modificada, calculada por um controlador lógico nebuloso (Fuzzy Logic Controller - FLC), reduz o tempo de convergência da rede.
- Os termos de restrições envolvidos com o problema a ser solucionado são confinados em um único termo de energia (E^{conf}), sem a necessidade de impor uma constante de ponderação associada.

- A inexistência das constantes de ponderação, escolhidas arbitrariamente, faz com que os pontos de equilíbrio da rede (solução final) sejam menos sensíveis à escolha do estado inicial.
- O termo de energia E^{ot} não interfere, durante o processamento, na minimização do termo de energia E^{conf} .
- Nenhum parâmetro de inicialização (controle) da rede é requerido para a execução da simulação.

Vale ressaltar ainda, que o mapeamento de problemas de otimização utilizando a rede de Hopfield modificada consiste simplesmente em determinar as matrizes T^{val} e T^{ot} , e os vetores s e i^{ot} associados a cada tipo de problema. Analisa-se, na próxima seção, o comportamento dinâmico da rede de Hopfield modificada.

2.5-) DINÂMICA DA REDE DE HOPFIELD MODIFICADA

Nesta seção, analisa-se a convergência da rede de Hopfield modificada cuja dinâmica de operação é implementada através dos passos (I)-(IV) apresentados anteriormente. Em particular, considera-se que a região de operação na qual o vetor v está contido é limitada pelo hipercubo definido pela função de ativação ‘rampa-simétrica’ (2.10). A equação nodal descrevendo o comportamento dinâmico desta rede é obtida a partir de (2.1) para $\eta = 0$ e $v(t) = u(t)$, ou seja:

$$\dot{v}_i(t) = \sum_{j=1}^N T_{ij} \cdot v_j(t) + i_i^b$$

A análise dinâmica da rede é portanto inicializada com a obtenção de uma equação para \dot{v}^{val} , componente de \dot{v} que pertence ao subespaço-válido.

Como na rede de Hopfield modificada, v é constantemente confinado ao subespaço-válido pela aplicação da operação (I), isto é $\{v = T^{val} \cdot v + s\}$, qualquer componente de \dot{v} ortogonal a \dot{v}^{val} é continuamente suprimida. Logo, a componente \dot{v}^{val} (não \dot{v}) caracteriza melhor a dinâmica global da rede. Assim:

$$\begin{aligned}
 \dot{\mathbf{v}}^{\text{val}} &= \mathbf{T}^{\text{val}} \cdot \dot{\mathbf{v}} = \mathbf{T}^{\text{val}} (\mathbf{T}^{\text{ot}} \cdot \mathbf{v} + \mathbf{i}^{\text{ot}}) \\
 &= \mathbf{T}^{\text{val}} (\mathbf{T}^{\text{ot}} (\mathbf{T}^{\text{val}} \cdot \mathbf{v} + \mathbf{s}) + \mathbf{i}^{\text{ot}}) \\
 &= \mathbf{T}^{\text{val}} \mathbf{T}^{\text{ot}} \mathbf{T}^{\text{val}} \cdot \mathbf{v} + \mathbf{T}^{\text{val}} (\mathbf{T}^{\text{ot}} \cdot \mathbf{s} + \mathbf{i}^{\text{ot}})
 \end{aligned} \tag{2.12}$$

Da equação (2.12), verifica-se que $\dot{\mathbf{v}}^{\text{val}}$ é composta de duas partes: um termo independente de \mathbf{v} , $\mathbf{T}^{\text{val}} (\mathbf{T}^{\text{ot}} \cdot \mathbf{s} + \mathbf{i}^{\text{ot}})$, e um termo que depende de \mathbf{v} , $\mathbf{T}^{\text{val}} \mathbf{T}^{\text{ot}} \mathbf{T}^{\text{val}} \mathbf{v}$. Para simplificar, estes termos passam a ser definidos por:

$$\mathbf{T}^{\text{val}} \mathbf{T}^{\text{ot}} \mathbf{T}^{\text{val}} = \mathbf{A} \tag{2.13}$$

$$\mathbf{T}^{\text{val}} (\mathbf{T}^{\text{ot}} \cdot \mathbf{s} + \mathbf{i}^{\text{ot}}) = \mathbf{b} \tag{2.14}$$

Assim, a equação (2.12) torna-se:

$$\dot{\mathbf{v}}^{\text{val}} = \mathbf{A} \cdot \mathbf{v} + \mathbf{b} = \mathbf{A} \cdot \mathbf{v}^{\text{val}} + \mathbf{b} \tag{2.15}$$

onde: $\mathbf{v}^{\text{val}} = \mathbf{T}^{\text{val}} \cdot \mathbf{v}$ e $\mathbf{T}^{\text{val}} \cdot \mathbf{T}^{\text{val}} = \mathbf{T}^{\text{val}}$.

No Apêndice B, mostra-se que com \mathbf{v} confinado no subespaço-válido (isto é: $\mathbf{v} = \mathbf{T}^{\text{val}} \cdot \mathbf{v} + \mathbf{s}$, e $\mathbf{T}^{\text{val}} \cdot \mathbf{s} = \mathbf{0}$), E^{ot} pode ser descrito por:

$$E^{\text{ot}} = -\frac{1}{2} \mathbf{v}^{\text{T}} \cdot \mathbf{A} \cdot \mathbf{v} - \mathbf{b}^{\text{T}} \cdot \mathbf{v} \tag{2.16}$$

Logo a dinâmica $\{\dot{\mathbf{v}} = \dot{\mathbf{v}}^{\text{val}} = \mathbf{A} \cdot \mathbf{v} + \mathbf{b}\}$ executa a minimização de E^{ot} dentro do subespaço-válido. Como resultado, uma solução válida que minimiza E^{ot} será obtida pela rede. Para sistemas invariantes no tempo (autônomos), a solução geral de (2.15) pode ser descrita por meio de uma equação exponencial matricial [Dazz75, Rose70, Vidy78]:

$$\mathbf{v}^{\text{val}}(t) = e^{\mathbf{A}t} \mathbf{v}_0^{\text{val}} + \int_0^t e^{\mathbf{A}(t-\tau)} \mathbf{b} \cdot d\tau \tag{2.17}$$

onde $\mathbf{v}_0^{\text{val}}$ é o valor de \mathbf{v}^{val} inicializado no tempo $\{t = 0\}$. Este valor é geralmente escolhido como um vetor com elementos aleatórios pequenos.

Dado que a expansão em série da exponencial $e^{\mathbf{A}\zeta}$ é dada por:

$$e^{\mathbf{A}\zeta} = \sum_{k=0}^{\infty} \frac{(\mathbf{A}\zeta)^k}{k!}$$

a equação (2.17) torna-se:

$$\begin{aligned}
 \mathbf{v}^{\text{val}}(t) &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \mathbf{A}^k \mathbf{v}_0^{\text{val}} + \int_0^t \sum_{k=0}^{\infty} \frac{(t-\tau)^k}{k!} \mathbf{A}^k \mathbf{b} \cdot d\tau \\
 &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \mathbf{A}^k \mathbf{v}_0^{\text{val}} + \sum_{k=0}^{\infty} \frac{\mathbf{A}^k \mathbf{b}}{k!} \int_0^t (t-\tau)^k \cdot d\tau \\
 &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \mathbf{A}^k \mathbf{v}_0^{\text{val}} + \sum_{k=0}^{\infty} \frac{\mathbf{A}^k \mathbf{b}}{k!} \left(\frac{t^{k+1}}{k+1} \right) \\
 &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \mathbf{A}^k \mathbf{v}_0^{\text{val}} + \sum_{k=0}^{\infty} \frac{t^{k+1}}{(k+1)!} \mathbf{A}^k \mathbf{b}
 \end{aligned} \tag{2.18}$$

Para se analisar o comportamento de \mathbf{v}^{val} durante o processo de convergência da rede, deve-se escrever os vetores \mathbf{v}^{val} , $\mathbf{v}_0^{\text{val}}$ e \mathbf{b} em termos de suas componentes expressas no espaço coordenado gerado pelos autovetores normalizados da matriz \mathbf{A} . Para isto, considere $\lambda_1, \lambda_2, \dots, \lambda_N$ os autovalores de \mathbf{A} , com autovetores normalizados $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N$. Para se distinguir entre os autovalores nulos e não nulos de \mathbf{A} , define-se o conjunto Z , tal que $\lambda_i = 0$ para $i \in Z$, e $\lambda_i \neq 0$ para $i \notin Z$. A decomposição de \mathbf{v}^{val} , $\mathbf{v}_0^{\text{val}}$ e \mathbf{b} na direção dos autovetores de \mathbf{A} gera:

$$\mathbf{v}^{\text{val}} = \sum_{i=1}^N v_i \mathbf{u}_i, \quad \mathbf{v}_0^{\text{val}} = \sum_{i=1}^N o_i \mathbf{u}_i, \quad \mathbf{b} = \sum_{i=1}^N b_i \mathbf{u}_i \tag{2.19}$$

onde v_i, o_i e b_i são, respectivamente, os valores da i -ésima componente dos vetores \mathbf{v}^{val} , $\mathbf{v}_0^{\text{val}}$ e \mathbf{b} , representadas no espaço coordenado gerado pelos autovetores de \mathbf{A} .

A partir de (2.19), obtém-se:

$$\mathbf{A}^k \mathbf{v}_0^{\text{val}} = \sum_{i=1}^N o_i \lambda_i^k \mathbf{u}_i, \quad \mathbf{A}^k \mathbf{b} = \sum_{i=1}^N b_i \lambda_i^k \mathbf{u}_i \tag{2.20}$$

Com a utilização de (2.20), a equação (2.18) torna-se:

$$\begin{aligned}
 \mathbf{v}^{\text{val}}(t) &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \sum_{i=1}^N o_i \lambda_i^k \mathbf{u}_i + \sum_{k=0}^{\infty} \frac{t^{k+1}}{(k+1)!} \sum_{i=1}^N b_i \lambda_i^k \mathbf{u}_i \\
 &= \sum_{i=1}^N o_i \mathbf{u}_i \underbrace{\sum_{k=0}^{\infty} \frac{t^k \lambda_i^k}{k!}}_{e^{\lambda_i t}} + \sum_{i \notin Z} \frac{b_i \mathbf{u}_i}{\lambda_i} \sum_{k=0}^{\infty} \frac{t^{k+1} \lambda_i^{k+1}}{(k+1)!} + \sum_{i \in Z} b_i \mathbf{u}_i \sum_{k=0}^{\infty} \frac{t^{k+1} 0^k}{(k+1)!} \\
 &= \sum_{i=1}^N e^{\lambda_i t} o_i \mathbf{u}_i + \sum_{i \notin Z} \frac{b_i \mathbf{u}_i}{\lambda_i} \left(\sum_{k=0}^{\infty} \frac{t^k \lambda_i^k}{k!} - 1 \right) + \sum_{i \in Z} b_i \mathbf{u}_i t \\
 &= \sum_{i=1}^N e^{\lambda_i t} o_i \mathbf{u}_i + \sum_{i \notin Z} \frac{b_i \mathbf{u}_i}{\lambda_i} (e^{\lambda_i t} - 1) + \sum_{i \in Z} b_i \mathbf{u}_i t
 \end{aligned} \tag{2.21}$$

A equação (2.21) é válida para valores arbitrários de \mathbf{A} , \mathbf{b} e $\mathbf{v}_0^{\text{val}}$. Entretanto, pode-se simplificar esta equação a partir da definição de \mathbf{A} e \mathbf{b} dada, nas equações (2.13) e (2.14). Desta análise, verifica-se que \mathbf{b} permanece integralmente no subespaço-válido durante toda convergência, de modo que $b_i = 0$ para $i \in Z$ (Esta análise é também realizada no Apêndice B). Assim, a equação (2.21) torna-se:

$$\mathbf{v}^{\text{val}}(t) = \sum_{i=1}^N e^{\lambda_i t} o_i \mathbf{u}_i + \sum_{i \notin Z} \frac{b_i \mathbf{u}_i}{\lambda_i} (e^{\lambda_i t} - 1) \tag{2.22}$$

Para ‘t’ pequeno, tem-se que:

$$e^{\lambda_i t} \approx 1 + \lambda_i t$$

Logo, a equação (2.22) torna-se:

$$\mathbf{v}^{\text{val}}(t) \approx \sum_{i=1}^N [o_i (1 + \lambda_i t) + b_i t] \cdot \mathbf{u}_i \tag{2.23}$$

No Apêndice B é mostrado que no início do processo de convergência (‘t’ pequeno) da rede tem-se que $\mathbf{b} = -\nabla E^{\text{ot}}$. Como $\mathbf{v}_0^{\text{val}}$ foi escolhido aleatoriamente pequeno, os termos o_i são pequenos em comparação com os b_i . Portanto, a equação (2.23) transforma-se em:

$$\mathbf{v}^{\text{val}}(t) \approx t \cdot \sum_{i=1}^N b_i \mathbf{u}_i = t \cdot \mathbf{b} \tag{2.24}$$

Verifica-se que os valores de \mathbf{v}^{val} inicialmente partem na direção do vetor \mathbf{b} . No limite, para ‘t’ grande, e visto que \mathbf{v} é limitado pelo hipercubo definido pela função de ativação rampa-simétrica dada em (2.10), a equação (2.22) indica que \mathbf{v}^{val} tenderá à direção dos

autovetores de A correspondente ao maior autovalor positivo, onde u^{\max} denota estes autovetores (e λ^{\max} o autovalor correspondente) [Vidy78]. Assim, b e u^{\max} são os vetores que mais influenciam a dinâmica de v^{val} . Quando v converge para um ponto de equilíbrio válido, o v^{val} associado contém invariavelmente uma componente significativa na direção de u^{\max} .

Mostra-se, assim, que a rede de Hopfield modificada, inicializada em um ponto aleatório, convergirá para um ponto de equilíbrio válido. Este ponto de equilíbrio está contido dentro do hiper-cubo definido pela função de ativação ‘rampa simétrica’.

2.6-) DESCRIÇÃO DO CONTROLADOR LÓGICO NEBULOSO

O passo de iteração da rede de Hopfield modificada Δt em (2.11) administra a velocidade de convergência da rede, bem como é responsável pela precisão dos vetores de equilíbrio. Semelhante ao algoritmo clássico de Gauss, deseja-se que no início este valor de passo seja grande e vá diminuindo a medida que a rede converge para seus pontos de equilíbrio. Também é desejável que o tamanho do passo seja independente do problema tratado. O tamanho do passo de iteração (Δt) da rede de Hopfield modificada é determinado por um controlador lógico nebuloso. Utiliza-se este controlador para minimizar o tempo de convergência da rede, sem prejuízo da precisão desejada para os valores dos pontos de equilíbrio.

A teoria de sistemas lógicos nebulosos, introduzida por Zadeh [Zade65], permite a manipulação de informações incertas. Em lógica nebulosa, um sistema pode representar conceitos imprecisos, não apenas por intermédio de variáveis linguísticas (por exemplo: grande, baixo e lento), mas também através de princípios matemáticos bem estabelecidos; além disso pode-se utilizar estes conceitos para realizar deduções sobre um sistema [Cox92].

A utilização mais significativa da teoria dos conjuntos nebulosos, em sistemas baseados em conhecimento, são os controladores lógicos nebulosos [Lee90]. Num controlador nebuloso baseado em regras, as variáveis de entrada são normalizadas e convertidas em representações nebulosas. As bases de regras são executadas em paralelo, produzindo conseqüentemente uma região nebulosa para cada variável. Em seguida, as regiões são convertidas em valores ‘crisp’ (não-nebuloso) com o objetivo de determinar uma ação de controle (não nebulosa) que melhor represente a decisão nebulosa.

Na teoria de conjuntos nebulosos, considera-se inicialmente um universo de discurso U , onde 'u' é um elemento genérico. Um conjunto nebuloso F , definido neste universo, é caracterizado por uma função de pertinência $\mu_F(\cdot)$ sobre o intervalo $[0,1]$. O grau de pertinência do elemento 'u' ao conjunto F é dado pelo valor $\mu_F(u)$. Assim, considerando dois conjuntos nebulosos A e B em U , com funções de pertinência μ_A e μ_B , respectivamente, as operações básicas de união, interseção e complemento, podem ser definidas através de suas funções de pertinência por:

$$\text{Uni\~ao: } \mu_{A \cup B} = \max\{ \mu_A(u), \mu_B(u) \} \quad (2.25)$$

$$\text{Interse\~c\~ao: } \mu_{A \cap B} = \min\{ \mu_A(u), \mu_B(u) \} \quad (2.26)$$

$$\text{Complemento: } \mu_{\bar{A}}(u) = 1 - \mu_A(u) \quad (2.27)$$

Um sistema de decis\~ao nebuloso, ou controlador nebuloso, \u00e9 ent\~ao formado por um conjunto de regras de controle, \u00e0s quais se associam implica\~oes nebulosas utilizando as opera\~oes precedentes. Uma regra nebulosa t\u00edpica 'Se (x \u00e9 A_i e y \u00e9 B_i) ent\~ao (z \u00e9 C_i)' \u00e9 implementada como uma implica\~ao (rela\~ao) nebulosa R_i , cuja fun\~ao de pertin\u00eancia μ_{R_i} pode ser definida por :

$$\begin{aligned} \mu_{R_i} &\stackrel{\Delta}{=} \mu(A_i \underline{e} B_i \rightarrow C_i)(u,v,w) \\ &= [\mu_{A_i}(u) \underline{e} \mu_{B_i}(v)] \rightarrow \mu_{C_i}(w) \end{aligned}$$

onde $A_i \underline{e} B_i$ \u00e9 um conjunto nebuloso $A_i \times B_i$ em $U \times V$; $R_i = (A_i \underline{e} B_i) \rightarrow C_i$ \u00e9 uma rela\~ao nebulosa em $U \times V \times W$; e o operador (\rightarrow) indica a fun\~ao de implica\~ao nebulosa. Considerando regras do tipo 'Se (u \u00e9 A_i) ent\~ao (v \u00e9 B_i)', a fun\~ao de pertin\u00eancia μ_{R_i} \u00e9 :

$$\mu_{R_i} = \min\{ \mu_{A_i}(u), \mu_{B_i}(v) \}; \quad u \in U, v \in V \quad (2.28)$$

Em um sistema nebuloso, cada regra nebulosa \u00e9 representada por uma rela\~ao nebulosa. Quando um sistema nebuloso \u00e9 caracterizado por um conjunto de regras, a rela\~ao nebulosa R , representando uma combina\~ao de todas as regras nebulosas R_i pertencentes ao sistema, \u00e9 definida atrav\u00e9s de um operador de agrega\~ao de regras:

$$R = \text{agreg}(R_1, R_2, \dots, R_n) \quad (2.29)$$

onde 'n' \u00e9 o n\u00famero de regras do sistema nebuloso. O operador de agrega\~ao \u00e9 interpretado, usualmente, como um operador de uni\~ao.

Utilizando-se a regra de inferência composicional de Zadeh [Zade73], dada a relação R, de U a V, e um conjunto nebuloso de U denotado por A'; o conjunto nebuloso B' de V, inferido a partir de A', tem função de pertinência definida por (procedimento max-min):

$$\mu_{B'}(v) = \max \{ \min (\mu_{A'}(u) , \mu_R(u,v)) \}; u \in U, v \in V \tag{2.30}$$

Baseado nestes conceitos e com o objetivo de priorizar o compromisso entre a eficiência de convergência da rede e a precisão dos valores dos pontos de equilíbrio da rede, associa-se à rede de Hopfield modificada, proposta na Seção 2.4, um controlador lógico nebuloso. As variáveis linguísticas de entrada do controlador são o erro quadrático médio (EQM) da saída entre dois instantes consecutivos, e a variação de energia do termo E^{ot} (2.6), dados por:

$$EQM(t) = \frac{1}{2} \sum_{i=1}^N [v_i(t) - v_i(t-1)]^2 \tag{2.31}$$

$$\Delta E^{ot}(t) = | E^{ot}(t) - E^{ot}(t-1) | \tag{2.32}$$

Ressalta-se que a equação (2.31) está fortemente relacionada com a minimização do termo de energia E^{conf} .

A saída do controlador lógico nebuloso é dada pela variável linguística Δt , que determina o passo de iteração da rede em (2.11). O universo de discurso associado a cada variável linguística de entrada tem valores no seguinte conjunto de termos: muito pequeno (MP), pequeno (PE), médio (ME), grande (GR), muito grande (MG). Para a variável de saída Δt , os valores são: pequeno (PE), médio (ME), grande (GR). Os valores de Δt são obtidos pela aplicação da regra de inferência composicional de Zadeh em regras do tipo ‘Se (EQM é grande) e (ΔE é pequeno) então (Δt é médio)’. A Tabela 2.1 sintetiza as regras utilizada pelo controlador.

		$\Delta E^{ot}(t)$				
		MP	PE	ME	GR	MG
EQM(t)	Δt					
	MP	PE	PE	ME	ME	ME
	PE	PE	PE	ME	ME	GR
	ME	ME	ME	ME	GR	GR
	GR	ME	ME	GR	GR	GR
MG	ME	GR	GR	GR	GR	

Tabela 2.1. Conjunto de Regras Nebulosas

Para cada regra, dada na Tabela 2.1, o grau de ativação da ação de controle é calculado de acordo com o resultado da combinação de seus antecedentes e consequentes. Todas as regras, com elementos tendo valores verdade não nulos, são ativadas. As funções de pertinência para as variáveis linguística do controlador são dadas na Figura 2.3. Os parâmetros ΔE_0 e E_{Q_0} , diferentes em cada tipo de problema a ser resolvido pela rede, são obtidos após as primeiras iterações executadas pela rede.

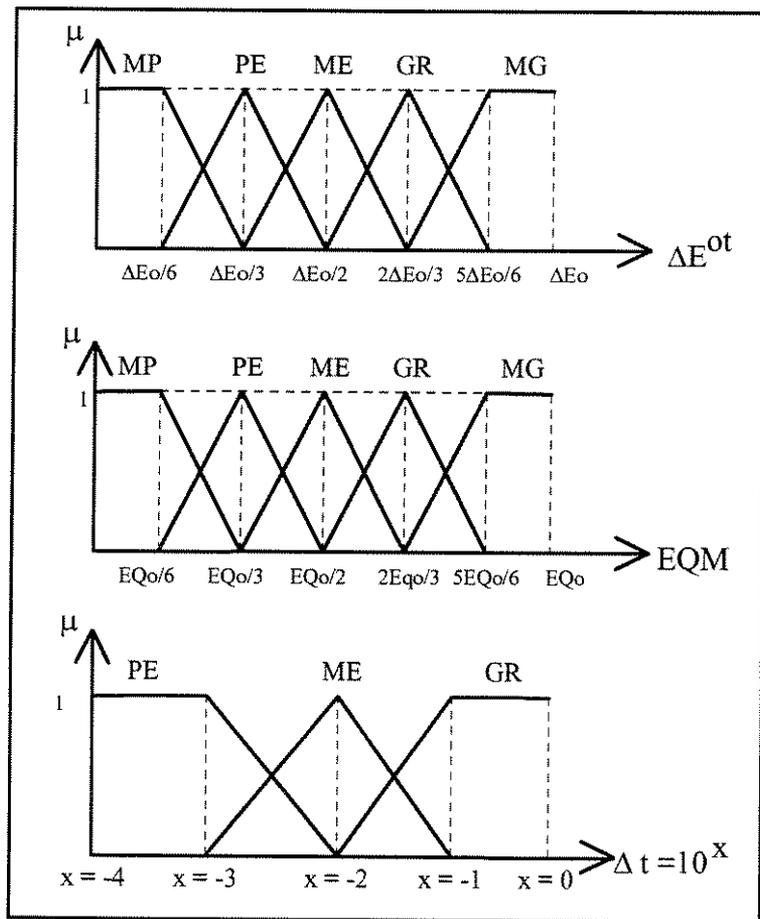


Figura 2.3. Funções de Pertinência do Controlador Nebuloso

Após a aplicação do procedimento de inferência, a ação de controle final consiste da agregação (união) de todas as contribuições proporcionadas por cada regra ativada [Lee90]. O valor ‘crisp’ (não nebuloso) de Δt é calculado pelo método do centro de área (CDA), dado por:

$$CDA = \frac{\sum_{k=1}^n \mu_{B'}(v_k) \cdot v_k}{\sum_{k=1}^n \mu_{B'}(v_k)} \quad (2.33)$$

onde 'n' é o número de níveis de quantização da saída, ou seja, o número de segmentos gerados pela discretização do universo de discurso.

2.7-) METODOLOGIA PARA MAPEAMENTO DE PROBLEMAS

Como discutido no Capítulo 1, uma das principais vantagens da rede de Hopfield modificada, proposta nesta tese, é sua capacidade de resolver vários tipos de problemas diferentes, utilizando uma arquitetura de rede única e uma mesma metodologia de mapeamento.

Em relação à arquitetura, a rede de Hopfield modificada é implementada como mostrado na Figura 2.2 e funciona segundo os passos (I)-(IV) descritos anteriormente. Já, a metodologia utilizada para o mapeamento de um determinado problema, que possa ser resolvido pela rede de Hopfield modificada, requer que o mesmo seja especificado (modelado) do seguinte modo:

- (i) A função objetivo do problema deve ser escrita na forma dada por (2.6), ou seja:

$$E^{ot}(t) = -\frac{1}{2} \mathbf{v}(t)^T \cdot \mathbf{T}^{ot} \cdot \mathbf{v}(t) - \mathbf{v}(t)^T \cdot \mathbf{i}^{ot}$$

- (ii) As restrições devem ser expressas por intermédio da equação do subespaço-válido definida em (2.8), ou seja:

$$\mathbf{v} = \mathbf{T}^{val} \cdot \mathbf{v} + \mathbf{s}$$

onde: $\mathbf{T}^{val} \cdot \mathbf{T}^{val} = \mathbf{T}^{val}$ e $\mathbf{T}^{val} \cdot \mathbf{s} = \mathbf{0}$.

- (iii) O domínio de variação de cada variável do problema deve ser definido através dos limites da função de ativação "rampa-simétrica" definida em (2.10), ou seja:

$$v_i \in [\lim_i^{inf}, \lim_i^{sup}].$$

Nos problemas de otimização linear e otimização não-linear, estes limites correspondem às variáveis canalizadas do problema. No caso dos problemas de otimização combinatorial e programação dinâmica, estes limites são especificados por zero e um ($\lim^{inf} = 0$ e $\lim^{sup} = 1$).

Nos próximos capítulos, utiliza-se a metodologia previamente descrita para mapear, através da rede de Hopfield modificada, alguns problemas de otimização e identificação, entre os quais destacam-se: otimização linear e não-linear, programação dinâmica e estimação paramétrica robusta.

2.8-) CONCLUSÃO

Este capítulo apresentou alguns fundamentos básicos envolvidos com RNA, e em particular com a rede de Hopfield convencional. A utilização desta rede, no mapeamento de problemas de otimização, requer que a função de energia associada à rede seja composta por vários termos de energia. Os principais problemas advindos desta forma de mapeamento são:

- Necessidade de uma escolha a priori dos parâmetros de ponderação dos termos associados às restrições.
- Interferência entre os termos de restrição e o termo de otimização, durante o processo de convergência da rede.
- A qualidade da solução final é afetada pelos valores dos parâmetros de ponderação.

Com as finalidades de contornar estes problemas inerentes à rede de Hopfield convencional, de aperfeiçoar a eficiência de simulações por computador e de fornecer uma nova metodologia para mapeamento de problemas de otimização, uma rede de Hopfield modificada foi desenvolvida. Esta rede utiliza a técnica do subespaço-válido para realizar a minimização da função de energia da rede e um controlador lógico nebuloso para calcular o tamanho do seu passo de iteração.

A técnica de subespaço-válido utilizada, aplicável a uma grande variedade de problemas de otimização, permite a derivação dos parâmetros internos da rede com o objetivo de incorporar de forma simples e compacta as várias restrições estruturais envolvidas no problema a ser mapeado. Esta técnica baseia-se na premissa de que todos os vetores de saídas da rede de Hopfield, que correspondem às soluções válidas de um problema específico, pertencem todos a um mesmo subespaço.

Por outro lado, o desempenho da rede de Hopfield convencional é afetado pelo uso de um passo de iteração (Δt) fixo. Para um Δt pequeno, a convergência para os pontos de equilíbrio é lenta, enquanto que para um Δt grande, a precisão destes pontos de equilíbrio é

prejudicada. Para a rede de Hopfield modificada, o passo de iteração é automaticamente calculado através de um controlador lógico nebuloso, cujos objetivos são:

- Acelerar a convergência da rede.
- Garantir a precisão dos pontos de equilíbrio.
- Tornar o cálculo do tamanho do passo independente do problema.

A operação da rede de Hopfield modificada é feita através de quatro passos principais, definidos por:

- (I) Projeção de \mathbf{v} no subespaço-válido (minimização indireta de E^{conf}).
- (II) Aplicação da função de ativação “rampa-simétrica”.
- (III) Cálculo do passo de iteração Δt pelo controlador lógico nebuloso.
- (IV) Minimização do termo de energia E^{ot} que corresponde à função objetivo do problema.

A partir destes quatro passos, e utilizando ferramentas de análise de sistema dinâmicos, foi possível estudar o comportamento da rede, bem como a sua convergência para os pontos de equilíbrio. Foi mostrado também que estes pontos de equilíbrio são sempre uma solução válida do problema mapeado.

Por fim, uma metodologia de mapeamento de problemas de otimização utilizando a rede de Hopfield modificada foi também proposta.

BIBLIOGRAFIA

- [Abe93] S. Abe. “Global convergence and suppression of spurious states of the Hopfield neural networks”. *IEEE Trans. on Circuits Syst. I*, vol. CAS-40, no. 4, pp. 246-257, 1993.
- [Aiye90] S. V. B. Aiyer, M. Niranja and F. Fallside. “A theoretical investigation into the performance of the Hopfield model”. *IEEE Transaction on Neural Networks*, vol. 1, no. 2, pp. 204-215, 1990.
- [Cox92] E. Cox. “Fuzzy fundamentals”. *IEEE Spectrum*, pp. 58-61, 1992.
- [Dazz75] J. J. D’azzo and C. H. Houppis. *Linear Control System Analysis and Design*. McGraw-Hill, New York, 1975.

- [Gee91] A. H. Gee and R. W. Prager. *Alternative energy functions for optimizing neural networks*. Technical Report Cued/F-Infeng/TR95, Cambridge University Department of Engineering, 1991.
- [Hayk94] S. Haykin. *Neural Networks - A Comprehensive Foundation*. McMillan Inc., Englewood Cliffs, NJ, 1994.
- [Hegd88] S. U. Hegde, J. L. Sweet and W. B. Levy. "Determination of parameters in a Hopfield/Tank computational network". *Proc. of International Conference on Neural Network*, 1988.
- [Hodg52] A. L. Hodgkin and L. F. Huxley. "A quantitative description of membrane current and its applications to conduction and excitation in nerve". *Journal of Physiology*, vol. 117, pp. 500-544, 1952.
- [Hopf84] J. J. Hopfield. "Neurons with graded response have collective computational properties like those of two-state neurons". *Proc. of National. Academy USA*, vol. 81, pp. 3088-3092, 1984.
- [Hopf85] J. J. Hopfield and D. W. Tank. "Neural computation of decisions in optimization problems". *Biological Cybernetics*, vol. 52, pp. 141-152, 1985.
- [Hush93] D. R. Hush and B. G. Horne. "Progress in supervised neural networks". *IEEE Signal Processing Magazine*, vol. 81, pp. 3088-3092, 1993.
- [Kamg90] B. Kamgar-Parsi and B. Kamgar-Parsi. "On problem solving with Hopfield neural networks". *Biological Cybernetics*, vol. 62, pp. 415-423, 1990.
- [Lee90] C. C. Lee. "Fuzzy logic in control systems: fuzzy logic controller, Part I and II". *IEEE Trans. on Systems, Man and Cybernetics*, vol. 20, pp. 404-435, 1990.
- [Li89] J. -H. Li, A. N. Michel and W. Porod. "Analysis and synthesis of a class of neural networks". *IEEE Trans. on Circuits Syst.*, vol. 36, pp. 1405-1422, 1989.
- [McCu43] W. S. McCulloch and W. Pitts. "A logical calculus of the ideas immanent in nervous activity". *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943.
- [Pete89] C. Peterson and B. Soderberg. "A new method for mapping optimization problems onto neural networks". *International Journal of Neural Networks*, vol. 1, no. 1, 1989.

[Rose70] H. H. Rosenbrook and C. Storey. *Mathematics of Dynamical Systems*. Thomas Nelson & Sons, London, 1970.

[Silv95] I. N. da Silva. *Estimação Paramétrica Robusta Através de Redes Neurais Artificiais*. Tese de Mestrado, DCA/FEE/UNICAMP, Campinas-SP, Janeiro de 1995.

[Vidy78] M. Vidyasagar. *Nonlinear Systems Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1978.

[Wils88] V. Wilson and G. S. Pawley. "On the stability of the TSP problem algorithm of Hopfield and Tank". *Biological Cybernetics*, vol. 58, pp. 63-70, 1988.

[Zade65] L. A. Zadeh. "Fuzzy sets". *Information and Control*, vol. 8, pp. 338-353, 1965.

[Zade73] L. A. Zadeh. "Outline of a new approach to the analysis of complex systems and decision processes". *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-3, pp. 28-44, 1973.

CAPÍTULO 3

OTIMIZAÇÃO LINEAR E QUADRÁTICA POR REDES NEURAS ARTIFICIAIS

3.1-) INTRODUÇÃO

A teoria de otimização possui um papel fundamental na matemática e na solução de problemas de Engenharia e Ciências. A filosofia envolvida com um problema de otimização está intrinsecamente ligada à seleção de valores admissíveis para um conjunto de variáveis interrelacionadas, considerando um critério de desempenho e/ou qualidade, relacionado ao problema a ser otimizado. Este critério é maximizado (ou minimizado) sujeito a um conjunto de restrições que limitam a seleção dos valores das variáveis envolvidas no problema.

Neste capítulo, aplica-se a metodologia proposta anteriormente na resolução de uma classe particular de problemas de otimização, ou seja, programação linear e quadrática. A fim de validar este novo método, faz-se também uma análise teórica da complexidade da rede de Hopfield modificada, e compara-se esta rede tanto em relação ao desempenho quanto à complexidade, com outras redes comumente usadas na solução de problemas de programação linear.

A organização do capítulo é a seguinte. Na Seção 3.2 descreve-se como problemas de programação linear podem ser mapeados e solucionados através da rede de Hopfield modificada. Uma análise de complexidade dos algoritmos neurais abordados neste capítulo é também realizada com o propósito de avaliar as principais redes utilizadas em programação

linear. Na Seção 3.3, a rede de Hopfield modificada é aplicada na resolução de problemas de programação quadrática.

3.2-) PROBLEMAS DE OTIMIZAÇÃO LINEAR

Otimização linear ou programação linear é um ramo muito importante da teoria de otimização devido a sua elevada aplicabilidade. Uma grande classe de problemas significativos em engenharia podem ser formulados como problemas de programação linear ou podem ser aproximados por tais problemas. Por exemplo, problemas de transporte, de produção, economia, planejamento, controle ótimo discreto podem ser convertidos em problemas de programação linear. O objetivo da programação linear está relacionado com a necessidade de minimizar ou maximizar uma função linear na presença de igualdades e/ou desigualdades lineares.

As aplicações de redes neurais artificiais têm mostrado que a ferramenta é promissora para resolver tais problemas eficientemente. Problemas de Programação Linear (PL) têm sido resolvidos por quatro modelos principais de RNA: o modelo de Tank e Hopfield [Tank86], o modelo de Kennedy e Chua [Kenn88], o de Rodríguez-Vázquez et al. [Rodr90] e o de Zak et al. [Zak95]. Estes modelos serão comparados com a rede de Hopfield modificada com o propósito de avaliar as topologias utilizadas e comparar os resultados obtidos por cada um dos métodos.

3.2.1-) Mapeamento de Problemas de Programação Linear Através da Rede de Hopfield Modificada

A formulação de problemas de programação linear em termos de RNA pode ser realizada através de dois métodos principais. O primeiro método é o desenvolvimento de uma função de energia apropriada (função de Lyapunov) tal que o estado de menor energia da rede corresponda às soluções desejadas. Através desta função de energia o problema de minimização é transformado em um conjunto de equações diferenciais ordinárias, e baseado nestas equações, pode-se projetar a RNA [Kenn88]. O segundo método consiste em desenvolver um conjunto de equações diferenciais ordinárias e, em seguida, encontrar uma função de Lyapunov tal que todas as trajetórias do sistema convirjam para algum ponto de equilíbrio que corresponda às soluções desejadas [Zhan92].

Neste trabalho, a rede de Hopfield modificada foi desenvolvida tendo por base o primeiro método descrito no parágrafo anterior. O problema de programação linear geral, com N-variáveis que podem ser limitadas ou não, e possuindo M^{eq} -restrições de igualdade e M^{in} -restrições de desigualdade pode ser descrito pelas seguintes equações:

$$\text{Minimizar } E^{ot}(\mathbf{v}) = \mathbf{c}^T \cdot \mathbf{v} \quad (3.1)$$

$$\text{sujeito a: } E^{conf}(\mathbf{v}) \begin{cases} (\mathbf{A}^{eq})^T \cdot \mathbf{v} = \mathbf{b}^{eq} \\ (\mathbf{A}^{in})^T \cdot \mathbf{v} \leq \mathbf{b}^{in} \end{cases} \quad (3.2)$$

$$\mathbf{z}^{min} \leq \mathbf{v} \leq \mathbf{z}^{max} \quad (3.4)$$

onde: $\mathbf{v}, \mathbf{c}, \mathbf{z}^{min}, \mathbf{z}^{max} \in \mathbb{R}^N$, $\mathbf{A}^{eq} \in \mathbb{R}^{N \times M^{eq}}$, $\mathbf{A}^{in} \in \mathbb{R}^{N \times M^{in}}$, $\mathbf{b}^{eq} \in \mathbb{R}^{M^{eq}}$ e $\mathbf{b}^{in} \in \mathbb{R}^{M^{in}}$. Na próxima subseção obtém-se as equações analíticas para os parâmetros internos da rede (definidos por \mathbf{T}^{val} , \mathbf{s} , \mathbf{T}^{ot} e \mathbf{i}^{ot}) utilizada para resolver problemas de programação linear.

Dedução das Equações Para \mathbf{T}^{val} , \mathbf{s} , \mathbf{T}^{ot} e \mathbf{i}^{ot}

Inicialmente, considera-se o caso em que existam apenas restrições de igualdade conforme dado em (3.2). As restrições de igualdade constituem um sistema de equações lineares que determinam um subespaço de soluções. O subespaço definido por \mathbf{v} pode ser descrito por:

$$\mathbf{v} = \mathbf{A}^{eq} \cdot (\mathbf{A}^{eqT} \cdot \mathbf{A}^{eq})^{-1} \cdot \mathbf{b}^{eq} \quad (3.5)$$

Se as linhas de \mathbf{A}^{eq} forem linearmente independentes ($\text{rank}(\mathbf{A}^{eq})=M^{eq}$), o vetor \mathbf{v} será único. Neste caso, a equação do subespaço-válido ($\mathbf{v} = \mathbf{T}^{val} \cdot \mathbf{v} + \mathbf{s}$) pode ser obtida igualando-se o vetor \mathbf{s} da equação do subespaço-válido ao respectivo valor de \mathbf{v} em (3.5). Esta atribuição significa que a equação do subespaço-válido contém a solução dada por (3.5), isto é:

$$\begin{aligned} \mathbf{v} &= \mathbf{T}^{val} \cdot \mathbf{v} + \mathbf{A}^{eq} \cdot (\mathbf{A}^{eqT} \cdot \mathbf{A}^{eq})^{-1} \cdot \mathbf{b}^{eq} \\ \mathbf{T}^{val} \cdot \mathbf{v} &= \mathbf{v} - \mathbf{A}^{eq} \cdot (\mathbf{A}^{eqT} \cdot \mathbf{A}^{eq})^{-1} \cdot \mathbf{b}^{eq} \end{aligned} \quad (3.6)$$

Utilizando-se a equação (3.2), obtém-se a partir de (3.6) a expressão para \mathbf{T}^{val} :

$$\begin{aligned} \mathbf{T}^{val} \cdot \mathbf{v} &= \mathbf{v} - \mathbf{A}^{eq} \cdot (\mathbf{A}^{eqT} \cdot \mathbf{A}^{eq})^{-1} \cdot \mathbf{A}^{eqT} \cdot \mathbf{v} \\ \mathbf{T}^{val} \cdot \mathbf{v} &= [\mathbf{I} - \mathbf{A}^{eq} \cdot (\mathbf{A}^{eqT} \cdot \mathbf{A}^{eq})^{-1} \cdot \mathbf{A}^{eqT}] \cdot \mathbf{v} \end{aligned}$$

Assim, os termos \mathbf{T}^{val} e \mathbf{s} do subespaço-válido para o problema de programação linear são dados por:

$$\mathbf{T}^{\text{val}} = \mathbf{I} - \mathbf{A}^{\text{eq}} \cdot (\mathbf{A}^{\text{eqT}} \cdot \mathbf{A}^{\text{eq}})^{-1} \cdot \mathbf{A}^{\text{eqT}} \quad (3.7)$$

$$\mathbf{s} = \mathbf{A}^{\text{eq}} \cdot (\mathbf{A}^{\text{eqT}} \cdot \mathbf{A}^{\text{eq}})^{-1} \cdot \mathbf{b}^{\text{eq}} \quad (3.8)$$

Estas equações obedecem as propriedades do subespaço-válido, ou seja, $(\mathbf{T}^{\text{val}} \cdot \mathbf{T}^{\text{val}} = \mathbf{T}^{\text{val}})$ e $(\mathbf{T}^{\text{val}} \cdot \mathbf{s} = \mathbf{0})$, portanto podem ser usadas como descrevendo este subespaço no passo (I) da arquitetura proposta na Seção 2.4. É importante ressaltar que as restrições de igualdade são combinadas em um único termo na função de energia da rede. Além disso, este termo não interfere na função objetivo do problema. A função de ativação dos neurônios, definida por (2.10) para a rede de Hopfield modificada, naturalmente faz que a condição (3.4) seja satisfeita.

A técnica descrita para as restrições de igualdade também pode ser estendida para as restrições de desigualdade na forma dada por (3.3). Neste caso, deve-se converter as restrições de desigualdade em restrições de igualdade introduzindo-se variáveis auxiliares (estendidas) para cada restrição de desigualdade. Assim, uma restrição de desigualdade típica dada por:

$$[\mathbf{A}^{\text{in}}]_{i1}^T \cdot v_1 + [\mathbf{A}^{\text{in}}]_{i2}^T \cdot v_2 + \dots + [\mathbf{A}^{\text{in}}]_{iN}^T \cdot v_N \leq b_i^{\text{in}} \quad (3.9)$$

torna-se:

$$[\mathbf{A}^{\text{in}}]_{i1}^T \cdot v_1 + [\mathbf{A}^{\text{in}}]_{i2}^T \cdot v_2 + \dots + [\mathbf{A}^{\text{in}}]_{iN}^T \cdot v_N + \sum_{j=1}^{M^{\text{in}}} q_j \cdot w_j = b_i^{\text{in}} \quad (3.10)$$

onde w_j são variáveis auxiliares que podem ser tratadas como componentes do vetor \mathbf{v} , e q_j são constantes dadas pela função impulso de Kronecker:

$$q_j = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} \quad (3.11)$$

Após a transformação do conjunto de restrições de desigualdade em um conjunto de restrições de igualdade, o vetor \mathbf{v}^+ que incorpora também as variáveis auxiliares é definido por $\mathbf{v}^{+T} = [\mathbf{v}^T \ \mathbf{w}^T]$, onde \mathbf{w}^T é o vetor de variáveis auxiliares $[w_1 \ w_2 \ \dots \ w_{M^{\text{in}}}]$. A dimensão associada ao vetor \mathbf{v}^+ é dada por $N^+ = N + M^{\text{in}}$, e a matriz \mathbf{A}^{+T} é de dimensão $(N+M^{\text{in}}) \times (M^{\text{eq}}+M^{\text{in}})$. Portanto, o problema de programação linear geral pode ser descrito por:

$$\text{Minimizar } E^{\text{ot}}(\mathbf{v}) = -\frac{1}{2} \mathbf{v}^{+\text{T}} \cdot \mathbf{T}^{\text{ot}^+} \cdot \mathbf{v}^+ - \mathbf{v}^{+\text{T}} \cdot \mathbf{i}^{\text{ot}^+} = \mathbf{c}^{+\text{T}} \cdot \mathbf{v}^+ \quad (3.12)$$

$$\text{sujeito a } E^{\text{rest}}(\mathbf{v}): \mathbf{A}^{+\text{T}} \cdot \mathbf{v}^+ = \mathbf{b}^+ \quad (3.13)$$

$$z_i^{\text{min}} \leq v_i^+ \leq z_i^{\text{max}} \quad i \in \{1..N\} \quad (3.14)$$

$$v_i^+ \geq 0 \quad i \in \{N+1..N^+\} \quad (3.15)$$

$$\text{onde: } \mathbf{A}^+ = \begin{bmatrix} \mathbf{A}^{\text{eq}} & \mathbf{A}^{\text{in}} \\ \mathbf{0} & \mathbf{I}^{\text{in}} \end{bmatrix}, \quad \mathbf{b}^+ = \begin{bmatrix} \mathbf{b}^{\text{eq}} \\ \mathbf{b}^{\text{in}} \end{bmatrix}, \quad \mathbf{v}^+ = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}, \quad \mathbf{c}^+ = [c_1 \ c_2 \ \dots \ c_N \ \underbrace{0 \ 0 \ \dots \ 0}_{M^{\text{in}}}]$$

Vale ressaltar que E^{ot} não depende das variáveis auxiliares \mathbf{w} , e a restrição de igualdade estendida (3.13) incorpora tanto as restrições de igualdade como as restrições de desigualdade originais (3.3). Além disso, a condição (3.4) fica dividida em duas partes: a condição (3.14) que define o intervalo de limitação para as variáveis originais v_i (básicas) pertencentes ao vetor \mathbf{v}^+ , e a condição (3.15) que estabelece que as variáveis estendidas w_i devem ser positivas.

Portanto, a matriz \mathbf{T}^{val} em (3.7) e o vetor \mathbf{s} em (3.8) tornam-se:

$$\mathbf{T}^{\text{val}} = \mathbf{I} - \mathbf{A}^+ \cdot (\mathbf{A}^{+\text{T}} \cdot \mathbf{A}^+)^{-1} \cdot \mathbf{A}^{+\text{T}} \quad (3.16)$$

$$\mathbf{s} = \mathbf{A}^+ \cdot (\mathbf{A}^{+\text{T}} \cdot \mathbf{A}^+)^{-1} \cdot \mathbf{b}^+ \quad (3.17)$$

onde a matriz $\mathbf{T}^{\text{val}} \in \mathcal{R}^{N^+ \times N^+}$ e o vetor $\mathbf{s} \in \mathcal{R}^{N^+}$. Os parâmetros \mathbf{T}^{ot^+} e \mathbf{i}^{ot^+} do termo de energia E^{ot} são dados por:

$$\mathbf{T}^{\text{ot}^+} = \begin{bmatrix} \mathbf{T}^{\text{ot}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (3.18)$$

$$\mathbf{i}^{\text{ot}^+} = \begin{bmatrix} \mathbf{i}^{\text{ot}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} -\mathbf{c} \\ \mathbf{0} \end{bmatrix} \quad (3.19)$$

Para problemas de programação linear, assume-se que a matriz \mathbf{T}^{ot} em (3.18) é nula. Assim, o vetor \mathbf{i}^{ot} em (3.19) corresponde ao vetor de coeficientes \mathbf{c} da função objetivo (3.1), ou seja, ($\mathbf{i}^{\text{ot}} = -\mathbf{c}$).

A fim de validar o método proposto, realiza-se na próxima seção uma análise de complexidade da rede de Hopfield modificada e de quatro redes existentes na literatura que são também usadas na solução de problemas PL: a rede de Tank e Hopfield [Tank86], a rede de

Kennedy e Chua [Kenn88], a rede de Rodríguez-Vázquez et al. [Rodr90] e a rede de Zak et al. [Zak95]. Estas redes são brevemente descritas no Apêndice C.

3.2.2-) Análise de Complexidade das Redes Neurais Artificiais

A análise de complexidade para diversos algoritmos neurais têm sido investigada por muitos pesquisadores [Bren81, Gima89, Zak95]. Mais especificamente, a análise da arquitetura de um algoritmo de redes neurais permite prever os recursos necessários para a implementação da rede, tais como, o número de portas lógicas necessárias e/ou quantidade de memória necessária. Neste capítulo, analisa-se a complexidade de cada neurônio, o número de neurônios utilizados, e a complexidade do modelo para cada arquitetura abordada. Como é bem conhecido na literatura, as redes neurais podem ser implementadas tanto em hardware como em software. Numa implementação em hardware, pode-se utilizar estas medidas para comparar e prever a complexidade de área (espaço) relativa para cada rede, independente da tecnologia utilizada. Numa implementação em tempo-discreto, que pode ser em hardware ou software (que é o caso da rede proposta), estas medidas fornecem subsídios para comparar e prever o esforço computacional em cada iteração da rede. Assim, analisa-se a complexidade da rede de Hopfield modificada em resolver problemas dados pelas equações (3.1) a (3.4), onde têm-se N variáveis, M^{eq} restrições de igualdade, e M^{in} restrições de desigualdade.

Em geral, descreve-se a complexidade neural e a complexidade do modelo em termos da função complexidade assintótica $\Theta(\cdot)$, que é definida por [Papa82]:

$$\Theta(g(n)) = \left\{ \begin{array}{l} f(n), \text{ se existirem constantes positivas } k_1 \text{ e } k_2 \text{ tal que,} \\ \text{para 'n' grande suficiente, } k_1 \cdot g(n) \leq f(n) \leq k_2 \cdot g(n) \end{array} \right\} \quad (3.20)$$

onde: 'n' é o tamanho da instância considerada, $f(n)$ é uma função que representa algum critério de medição da eficiência dos algoritmos, e $g(n)$ é uma função que indica a ordem de complexidade de $f(n)$, isto é, $g(n)$ mede a taxa de crescimento de $f(n)$ em relação ao tamanho da instância. Assim, dadas duas funções $f(n)$ e $g(n)$, diz-se que $f(n)$ tem uma complexidade assintótica $g(n)$ ou que $f(n)$ é da ordem $\Theta(g(n))$.

Como um exemplo ilustrativo, admitindo-se que o critério de eficiência de um determinado algoritmo possa ser representada por $f(n) = n^2 + 100n$, então $f(n)$ tem ordem de complexidade $\Theta(n^2)$. Isto significa que quando 'n' for elevado, o valor da função $f(n)$ será

devido sobretudo à parcela n^2 , visto que a contribuição da parcela $(100n)$ torna-se quase insignificante. Formalmente, para este exemplo, tem-se:

$$\Theta(n^2) = \begin{cases} n^2 + 100n, \text{ tal que, } n^2 \leq n^2 + 100n \leq 2n^2, \\ \text{para todo } n \geq 100. \end{cases}$$

Neste caso, os valores das constantes k_1 e k_2 , referentes à (3.20), são dados por $k_1=1$ e $k_2=2$. Deve-se ressaltar que o número de iterações necessárias para a convergência das redes neurais não interfere na análise de complexidade das mesmas. Um estudo mais detalhado sobre teoria de complexidade de algoritmos pode ser encontrado em [Aho74, Gare79, Lewi81, Papa82].

Complexidade Neural

A complexidade de cada neurônio é definida como o número de multiplicações/divisões e adições/subtrações por iteração executado por cada neurônio. A distinção entre operações de multiplicação/divisão e adição/subtração deve-se ao custo de implementação em software e/ou hardware para cada uma desta classe pode ser significativamente diferente.

Para a rede de Hopfield modificada, proposta neste capítulo para solução de problemas de programação linear, a equação que sintetiza a operação de cada neurônio é dada por:

$$v_k^+ = \Delta t \cdot i^{ot^+} + (T_k^{val} \cdot v^+ + s_k) \tag{3.21}$$

onde T_k^{val} corresponde à k -ésima linha de T^{val} .

A complexidade neural (por neurônio) para a RHM é portanto $(1 + (M^{eq} + M^{in}))$ para as multiplicações e $(2 + (M^{eq} + M^{in}))$ para as adições/subtrações. Assim, a complexidade assintótica para as multiplicações e adições/subtrações dos neurônios é $\Theta(M^{eq} + M^{in})$.

Complexidade do Modelo

A complexidade do modelo é definida como o número total de multiplicações/divisões e adições/subtrações executada por iteração pela rede. O número de multiplicações/divisões e adições/subtrações são dados por:

- Número total de multiplicações/divisões = (número de multiplicações/divisões por neurônio) x (número de neurônios da rede) por iteração.

- Número total de adições/subtrações = (número de adições/subtrações por neurônio) x (número de neurônios da rede) por iteração.

Esta estimativa da complexidade do modelo é prática tanto para implementações em software como para implementações em hardware. Na implementação utilizando, por exemplo, circuito analógico, a complexidade de área está relacionada ao número total de multiplicações/divisões e adições/subtrações executadas em cada unidade processadora. Em simulações em tempo-discreto utilizando computador, a complexidade de tempo da iteração está também relacionada com o número total de multiplicações/divisões e adições/subtrações necessárias em cada algoritmo.

O número total de neurônios utilizados na RHM proposta é $(N + M^{in})$ neurônios. Logo, a complexidade do modelo é $(N + M^{in}).(1 + M^{eq} + M^{in})$ para as multiplicações, e $(N + M^{in}).(2 + M^{eq} + M^{in})$ para as adições/subtrações. Utilizando a definição (3.20), a complexidade assintótica do modelo para a rede de Hopfield modificada vale $\Theta((N + M^{in}).(M^{eq} + M^{in}))$ tanto para as multiplicações como para as adições/subtrações.

Outro aspecto de complexidade de redes neurais artificiais refere-se aos parâmetros intrínsecos de cada arquitetura, como eles devem ser inicializados e como esta inicialização influi no desempenho final da rede. No caso da RHM existe apenas um parâmetro intrínseco, o passo de iteração Δt , mas a sua inicialização fica a cargo do controlador nebuloso e, portanto, é transparente ao usuário e independente do problema. Pode-se portanto, considerar que para a RHM não existe nenhum parâmetro a ser inicializado.

Modelo de RNA	Número de Neurônios	Complex. Ass. Neurônio	Complex. Ass. Modelo	Parâmetros de Inicialização
Tank	N	$\Theta(M^{eq} + M^{in})$	$\Theta(N(M^{eq} + M^{in}))$	Δt
Kennedy	N	$\Theta(M^{eq} + M^{in})$	$\Theta(N(M^{eq} + M^{in}))$	s e Δt
Rodríguez	N	$\Theta(M^{eq} + M^{in})$	$\Theta(N(M^{eq} + M^{in}))$	s e Δt
Zak	$N + M^{in}$	$\Theta(M^{eq} + M^{in})$	$\Theta((N + M^{in}).(N + M^{eq} + M^{in}))$	ρ, γ e Δt
RHM	$N + M^{in}$	$\Theta(M^{eq} + M^{in})$	$\Theta((N + M^{in}).(M^{eq} + M^{in}))$	-

Tabela 3.1. Resultados da Análise de Complexidade dos Algoritmos (PL)

Utilizando-se os conceitos de complexidade, analisa-se no Apêndice C, para cada uma das quatro redes citadas anteriormente, a complexidade em resolver problemas dados pelas equações (3.1) a (3.4), onde têm-se N-variáveis, M^{eq} -restrições de igualdade, e M^{in} -restrições

de desigualdade. Os resultados desta análise se encontram ilustrados na Tabela 3.1. Para um problema com um número elevado de variáveis e restrições (isto é, $N \approx M^{\text{in}}, M^{\text{eq}}$), a ordem da complexidade do modelo é $\Theta(N^2)$ para todas as redes. Para um problema com um número elevado de variáveis comparado com o número total de restrições, a ordem de complexidade do modelo é $\Theta(N^2)$ para a rede de Zak et al. e para os outros modelos (Tank, Kennedy, Rodríguez e RHM) é $\Theta(N)$.

Ainda analisando a Tabela 3.1, na coluna intitulada ‘Parâmetros de Inicialização’ encontram-se os parâmetros que devem ser fornecidos (ajustados) no instante da inicialização das redes (s , ρ e γ : constantes de ponderação). Na rede de Kennedy e Chua [Kenn88], e Rodríguez-Vázquez et al. [Rodr90] é necessário o ajuste de dois parâmetros (s e Δt), enquanto que na rede de Zak et al. [Zak95] é necessário o ajuste de três parâmetros de inicialização (ρ , γ e Δt). A dificuldade ou a inexistência de procedimentos formais para a especificação destes parâmetros de inicialização pode tornar a rede inviável. Por outro lado, a rede de Tank e Hopfield [Tank86] tem a vantagem de ser necessário fornecer apenas o parâmetro Δt para a sua inicialização. Este parâmetro, para a RHM, é ajustado automaticamente pelo controlador lógico nebuloso.

3.2.3-) Exemplo de Simulação

Para ilustrar o desempenho da rede proposta, esta rede é utilizada na solução do problema ‘Gasoline Blending Problem’, proposto por [Ozan86] que contém restrições de igualdade e desigualdade. Os resultados obtidos utilizando esta rede são comparados aos de outras redes, citadas anteriormente, e que também foram aplicadas ao mesmo problema.

Problema de Composição de Gasolina

Uma companhia de óleo produz dois tipos de óleo bruto com as especificações dadas na Tabela 3.2. A companhia mistura o óleo de tipo 1 com o óleo de tipo 2 para a produção de gasolina de qualidade A e gasolina de qualidade B, com os seguintes requisitos:

- O número mínimo de octanos aceitável para a gasolina de qualidade A é 105. O número de octanos é estimado utilizando a média ponderada do mínimo de octanos de ambos óleos componentes.

- A pressão máxima de vapor aceitável para a gasolina de qualidade A é 8 lb/in². A pressão de vapor é estimada utilizando a média ponderada da pressão de vapor dos óleos componentes.
- Para a fabricação de gasolina com qualidade B não há nenhuma restrição em relação a mistura.

Após a produção de gasolina de qualidade A, todo óleo restante é misturado para produzir gasolina de qualidade B. A gasolina de qualidade A é vendida com um lucro de 8/unidade, e a gasolina de qualidade B com um lucro de 5/unidade. A produção diária de óleo disponível, para a fabricação das gasolinas, é também fornecida na Tabela 3.2.

O objetivo do problema consiste na determinação das quantidades de gasolina de qualidade A e B que devem ser produzidas diariamente, de modo a se obter o maior lucro possível. Este problema pode ser modelado como segue:

$$\begin{aligned}
 &\text{Maximizar } E^{\text{ol}}(\mathbf{x}): 8x_1 + 8x_2 + 5x_3 + 5x_4 \\
 &\text{sujeito a } E^{\text{conf}}(\mathbf{x}): \frac{110x_1 + 100x_2}{x_1 + x_2} \geq 105 \\
 &\qquad\qquad\qquad \frac{10x_1 + 5x_2}{x_1 + x_2} \leq 8 \\
 &\qquad\qquad\qquad x_1 + x_3 = 40 \\
 &\qquad\qquad\qquad x_2 + x_4 = 60 \\
 &\qquad\qquad\qquad x_1, x_2, x_3, x_4 \geq 0
 \end{aligned}$$

onde: x_1 = número de unidades de óleo tipo 1 na gasolina de qualidade A.
 x_2 = número de unidades de óleo tipo 2 na gasolina de qualidade A.
 x_3 = número de unidades de óleo tipo 1 na gasolina de qualidade B.
 x_4 = número de unidades de óleo tipo 2 na gasolina de qualidade B.

Óleo	Desempenho (No. Octanos)	Pressão de vapor (Lb/in ²)	Produção (un./dia)
1	110	10	40
2	100	5	60

Tabela 3.2. Especificação de Dados do Problema da Gasolina

A solução exata para este problema é dada por $\mathbf{x}^* = [40 \ 40 \ 0 \ 20]^T$, com $E^{\text{ol}}(\mathbf{x}^*) = 740$. Na Tabela 3.3, mostra-se uma comparação dos resultados obtidos por 5 redes

diferentes, que podem ser utilizadas na solução de problemas de programação linear [Maa92]: a rede de Tank e Hopfield [Tank86], a rede de Kennedy e Chua [Ken88], a rede de Rodríguez-Vázquez et al. [Rodr90], a rede de Zak et al. [Zak95] e a rede de Hopfield modificada (RHM) proposta na seção anterior. Verifica-se que a RHM apresenta resultados semelhantes aos obtidos pela rede proposta por Zak, tendo resultados ligeiramente melhor que os resultados obtidos pelas redes propostas por Tank, Kennedy, e Rodríguez.

RNA	x_1	x_2	x_3	x_4	E^{ot}
Tank	40.529	40.418	-0.014	19.564	745.326
Kennedy	40.193	40.132	-0.051	19.951	742.100
Rodríguez	41.702	41.451	-0.006	18.800	758.924
Zak	40.001	40.001	0.000	19.973	739.881
RHM	40.001	40.001	0.000	19.998	740.006

Tabela 3.3. Comparação dos Resultados (Problema da Gasolina)

A Tabela 3.4 mostra a importância do controlador lógico nebuloso associado à rede de Hopfield modificada. Nesta tabela, compara-se os resultados obtidos pela rede proposta quando se atribuem valores fixos para Δt , isto é, sem a utilização do controlador nebuloso. A coluna ‘Iterações’ indica o número de iterações necessárias para a convergência da rede. A partir da análise dos dados, para se ter resultados com a mesma precisão obtida quando se utiliza o controlador nebuloso, são necessárias cerca de 15000 iterações. Com o controlador nebuloso, a rede converge em cerca de 1300 iterações.

RHM	x_1	x_2	x_3	x_4	E^{ot}	Iterações
$\Delta t = 0.190$	41.637	41.642	-0.017	18.546	758.877	696
$\Delta t = 0.100$	40.871	40.824	-0.009	19.115	749.090	3149
$\Delta t = 0.010$	40.159	40.098	0.000	19.816	741.136	6925
$\Delta t = 0.001$	40.006	40.010	0.000	19.992	740.088	15512

Tabela 3.4. Comparação dos Resultados Para Δt Fixo

A Figura 3.1 ilustra a evolução dos valores de x_1 , x_2 , x_3 e x_4 obtidos pela RHM, com Δt calculado pelo controlador lógico nebuloso, em relação ao número de iterações. Os valores de ΔE_0 e E_{Q_0} , obtidos após as primeiras iterações, são ($\Delta E_0 = 3.832$) e ($E_{Q_0} = 0.129$). A Figura 3.2 mostra o comportamento da função objetivo do problema $E^{ot}(x)$, representada por $E(x)$, em função do número de iterações.

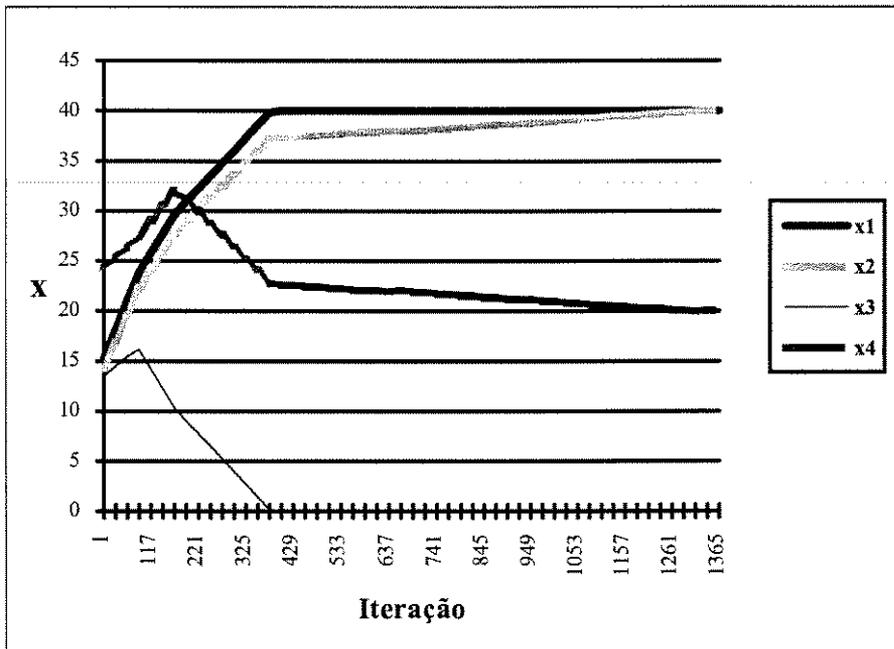


Figura 3.1. Evolução das Variáveis Para o Problema da Gasolina

A RHM foi testada também, considerando diversos valores iniciais, gerados aleatoriamente entre zero e um, atribuídos aos elementos do vetor x . Em todas as simulações a rede é globalmente convergente, sempre tendendo para a mesma solução, em um número de iterações de mesma ordem de grandeza. As restrições de não-negatividade foram mapeadas diretamente pela função de ativação rampa-simétrica, definida em (2.10), com $\lim_i^{\text{inf}} = 0$ e $\lim_i^{\text{sup}} = 1000$.

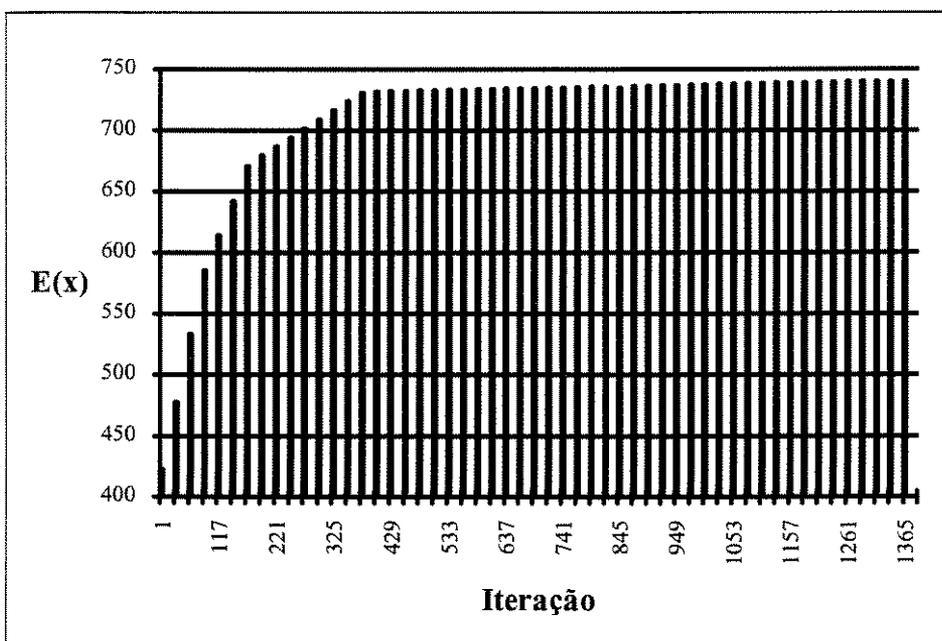


Figura 3.2. Comportamento da Função Objetivo Para o Problema da Gasolina

Na próxima subseção, destaca-se algumas considerações sobre a aplicação da rede de Hopfield modificada na solução de problemas de programação linear.

3.2.4-) Considerações Sobre a Aplicação da RHM na Solução de Problemas de Programação Linear

Da análise comparativa de complexidade e dos resultados do exemplo de simulação apresentados, constata-se que a RHM assemelha-se às outras redes em nível de complexidade e além disso, por intermédio do controlador nebuloso, tem-se um mecanismo eficiente para conduzir a sua convergência. A RHM possui ainda a vantagem de não ser necessário escolher nenhum parâmetro de ponderação para sua inicialização. Deste ponto de vista, o método proposto é inovador e eficiente. Para as outras redes, uma escolha inadequada destes parâmetros pode resultar em soluções infactíveis e/ou um maior tempo de convergência.

Deve-se ressaltar ainda que quando o problema de programação linear não estipula os valores de z^{\min} e z^{\max} em (3.4), referentes às variáveis canalizadas, as mesmas devem ser inicializadas com valores relativamente elevados tal que a solução do problema esteja entre z^{\min} e z^{\max} .

3.3-) PROBLEMAS DE OTIMIZAÇÃO QUADRÁTICA

Nesta seção desenvolve-se o mapeamento de problemas de otimização quadrática utilizando a rede de Hopfield modificada. Como no caso de programação linear, calcula-se as equações decrevendo o subespaço-válido e as matrizes T^{ot} e i^{ot} compondo a função objetivo do problema. Realiza-se também uma análise comparativa, através de resultados simulados, com a rede de Kennedy [Kenn88], que entre as topologias citadas anteriormente, é a única capaz de resolver problemas de programação quadrática.

3.3.1-) Mapeamento de Problemas de Programação Quadrática Através da Rede de Hopfield Modificada

Problemas de otimização quadrática ou programação quadrática (PQ) se referem, convencionalmente, ao problema de minimizar (ou maximizar) uma função quadrática sujeita a um conjunto de restrições lineares [Maa92]. Assim, o problema de programação quadrática geral com variáveis limitadas pode ser descrito pelas seguintes equações:

$$\text{Minimizar } E^{\text{ot}}(\mathbf{v}) = \frac{1}{2} \mathbf{v}^T \cdot \mathbf{Q} \cdot \mathbf{v} + \mathbf{v}^T \cdot \mathbf{c} = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N Q_{ji} \cdot v_i \cdot v_j + \sum_{i=1}^N v_i \cdot c_i \quad (3.22)$$

$$\text{sujeito a: } E^{\text{conf}}(\mathbf{v}) \begin{cases} (\mathbf{A}^{\text{eq}})^T \cdot \mathbf{v} = \mathbf{b}^{\text{eq}} \\ (\mathbf{A}^{\text{in}})^T \cdot \mathbf{v} \leq \mathbf{b}^{\text{in}} \end{cases} \quad (3.23)$$

$$\mathbf{z}^{\text{min}} \leq \mathbf{v} \leq \mathbf{z}^{\text{max}} \quad (3.25)$$

onde $\mathbf{c} \in \mathbb{R}^N$, $\mathbf{Q} \in \mathbb{R}^{N \times N}$ é uma matriz simétrica semidefinida positiva, e as equações (3.23), (3.24) e (3.25) são idênticas àquelas definidas para o problema de programação linear.

Dedução das Equações Para \mathbf{T}^{val} , \mathbf{s} , \mathbf{T}^{ot} e \mathbf{i}^{ot}

Desde que as restrições de igualdade e desigualdade impostas pelo problema de programação quadrática sejam idênticas às definidas para o problema de programação linear, os parâmetros \mathbf{T}^{val} e \mathbf{s} do subespaço-válido (que minimizam E^{conf}) serão os mesmos fornecidos pelas equações (3.16) e (3.17), ou seja:

$$\mathbf{T}^{\text{val}} = \mathbf{I} - \mathbf{A}^+ \cdot (\mathbf{A}^{+T} \cdot \mathbf{A}^+)^{-1} \cdot \mathbf{A}^{+T}$$

$$\mathbf{s} = \mathbf{A}^+ \cdot (\mathbf{A}^{+T} \cdot \mathbf{A}^+)^{-1} \cdot \mathbf{b}^+$$

No capítulo 2, mostrou-se que a minimização do termo de energia E^{ot} da rede de Hopfield modificada é efetuada alterando-se o movimento do vetor \mathbf{v} na direção oposta ao gradiente do termo de energia E^{ot} . Assim, igualando-se a equação (3.22) à equação (2.6), os parâmetros \mathbf{T}^{ot} e \mathbf{i}^{ot} {referentes às equações (3.18) e (3.19)} da RHM podem ser definidos por:

$$\mathbf{T}^{\text{ot}} = -\mathbf{Q} \quad (3.26)$$

$$\mathbf{i}^{\text{ot}} = -\mathbf{c} \quad (3.27)$$

Quando a matriz \mathbf{Q} e o vetor \mathbf{c} não estiverem prontamente acessíveis, os parâmetros \mathbf{T}^{ot} e \mathbf{i}^{ot} poderão ser calculados como:

$$\mathbf{T}^{\text{ot}} = \mathbf{0} \quad (3.28)$$

$$\mathbf{i}^{\text{ot}} = - \left[\frac{\partial E^{\text{ot}}(\mathbf{v})}{\partial v_1}, \frac{\partial E^{\text{ot}}(\mathbf{v})}{\partial v_2}, \dots, \frac{\partial E^{\text{ot}}(\mathbf{v})}{\partial v_N} \right] \quad (3.29)$$

3.3.2-) Resultados de Simulação

A rede de Hopfield modificada foi aplicada na solução do seguinte problema de programação quadrática [Kenn88]:

$$\text{Minimizar } E^{ot}(\mathbf{v}) = 0.4v_1 + \frac{1}{2}(5v_1^2 + 8v_2^2 + 4v_3^2) - 3v_1.v_2 - 3v_2.v_3$$

$$\text{sujeito a: } v_1 + v_2 + v_3 \geq 1$$

$$v_1, v_2, v_3 \geq 0$$

Os resultados teóricos para este problema são dados por:

$$v_1 = 0.2520, v_2 = 0.3328, v_3 = 0.4150, E^{ot} = 0.3811$$

RNA	v_1	v_2	v_3	E^{ot}
Kennedy	0.2570 (1.98%)	0.3320 (0.24%)	0.4120 (0.72%)	0.3820 (0.24%)
RHM	0.2520 (0.00%)	0.3329 (0.03%)	0.4151 (0.02%)	0.3812 (0.03%)

Tabela 3.5. Comparação dos Resultados (PQ)

Para problemas de programação quadrática, entre as redes analisadas neste capítulo, apenas a RHM e a rede de Kennedy e Chua podem ser utilizadas diretamente tanto em programação linear como em programação quadrática. Na Tabela 3.5, mostra-se uma comparação dos resultados obtidos com a RHM e a rede de Kennedy e Chua [Kenn88]. Nesta tabela, os valores entre parênteses denotam o erro relativo da respectiva variável em relação aos valores teóricos. O erro relativo é definido por:

$$\text{Erro}_i^{rel} = \frac{(v_i^{real} - v_i^{calc}).100}{v_i^{real}}\% \tag{3.30}$$

onde v_i^{real} é o valor teórico (verdadeiro) para cada variável do problema, e v_i^{calc} é o valor obtido pela rede neural considerada. Para este problema de programação quadrática, o maior erro relativo da RHM é 0.03%, enquanto que para a rede de Kennedy é 1.98%. As restrições de não-negatividade foram mapeadas diretamente pela função de ativação rampa-simétrica, definida em (2.10), com $\lim_i^{inf} = 0$ e $\lim_i^{sup} = 100$.

A Figura 3.3 apresenta a evolução dos valores de v_1, v_2 e v_3 obtidos pela RHM em relação ao número de iterações. Os valores de ΔE_0 e EQ_0 do controlador nebuloso, obtidos

após as primeiras iterações da rede, são ($\Delta E_0 = 1.625$) e ($EQ_0 = 0.092$). A Figura 3.4 ilustra o comportamento da função objetivo do problema em função do número de iterações.

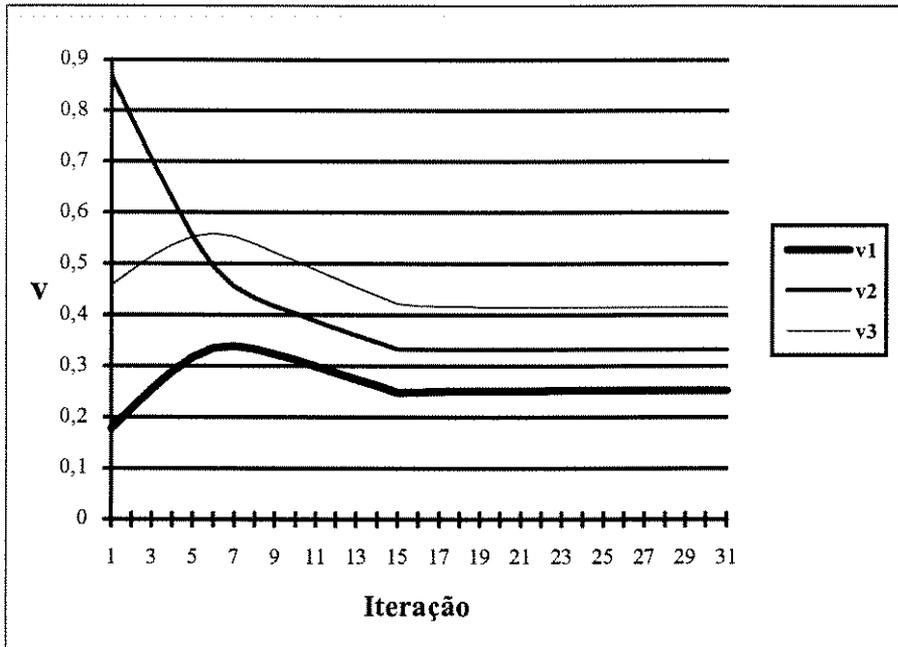


Figura 3.3. Evolução das Variáveis Para o Prob. de Prog. Quadrática

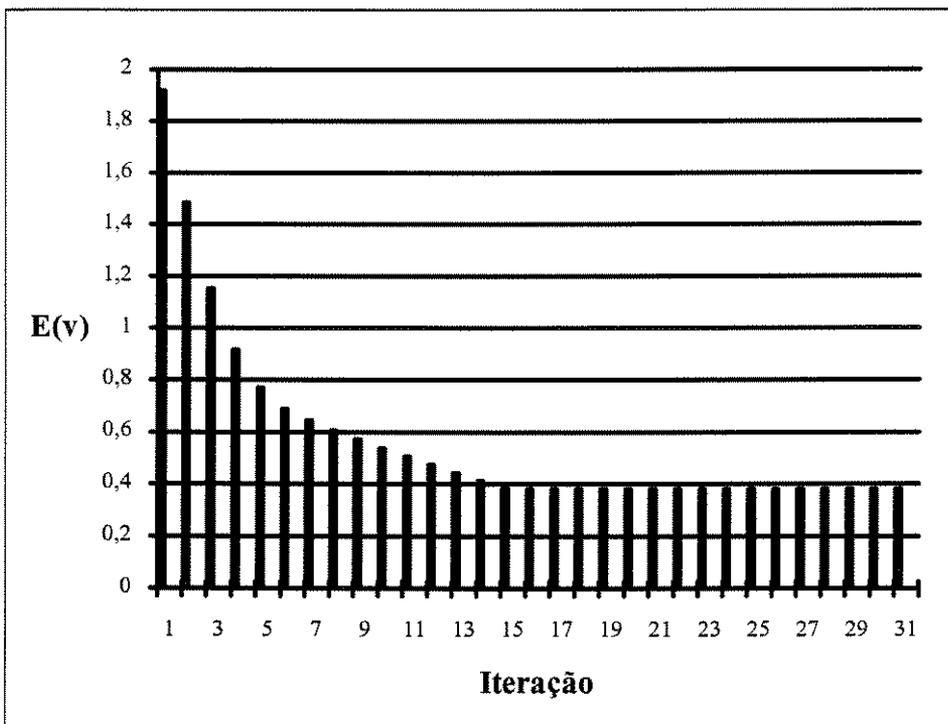


Figura 3.4. Comportamento da Função Objetivo Para o Prob. de Prog. Quadrática

A rede foi testada também, considerando diversos valores iniciais, gerados aleatoriamente entre zero e um, atribuídos aos elementos do vetor v . Em todas as simulações executadas, a rede sempre convergiu para a mesma solução, comprovando o que se afirma no Capítulo 2, que uma vez que as restrições são satisfeitas (por intermédio do subespaço-válido), a convergência da rede independe do valor inicial atribuído ao vetor v .

3.3.3-) Considerações Sobre a Aplicação da RHM na Solução de Problemas de Programação Quadrática

Para os problemas de programação quadrática, a equação que sintetiza a operação de cada neurônio da RHM é dada por:

$$v_k^+ = \Delta t. \left(\mathbf{T}_k^{ot^+} \cdot v^+ + i^{ot^+} \right) + \left(\mathbf{T}_k^{val} \cdot v^+ + s_k \right) \tag{3.31}$$

onde: $\mathbf{T}_k^{ot^+}$ corresponde à k -ésima linha de \mathbf{T}^{ot^+} .

\mathbf{T}_k^{val} corresponde à k -ésima linha de \mathbf{T}^{val} .

A complexidade neural (por neurônio) é portanto $(1 + 2(M^{eq} + M^{in}))$ para as multiplicações, e $(3 + 2(M^{eq} + M^{in}))$ para as adições/subtrações. Logo, a complexidade assintótica para as multiplicações e adições/subtrações é $\Theta(M^{eq} + M^{in})$.

Como o número de neurônios utilizados na RHM é $(N + M^{in})$, a complexidade do modelo é $(N + M^{in}).(1 + 2(M^{eq} + M^{in}))$ para as multiplicações, e $(N + M^{in}).(3 + 2(M^{eq} + M^{in}))$ para as adições/subtrações. A complexidade assintótica do modelo para a RHM vale $\Theta((N + M^{in}).(M^{eq} + M^{in}))$ tanto para as multiplicações como para as adições/subtrações.

Modelo de RNA	Número de Neurônios	Complex. Ass. Neurônio	Complex. Ass. Modelo	Parâmetros de Inicialização
Kennedy	N	$\Theta(M^{eq} + M^{in})$	$\Theta(N(M^{eq} + M^{in}))$	s e Δt
RHM	$N + M^{in}$	$\Theta(M^{eq} + M^{in})$	$\Theta((N + M^{in}).(M^{eq} + M^{in}))$	-

Tabela 3.6. Resultados da Análise de Complexidade dos Algoritmos (PQ)

A Tabela 3.6 apresenta o valores da complexidade assintótica da RHM e também da rede de Kennedy e Chua [Kenn88]. Verifica-se para ambas as redes que a complexidade neural e a complexidade do modelo são idênticas àquelas encontradas para problemas de programação linear. Como no caso linear, na RHM não é necessário o ajuste de nenhum parâmetro, enquanto que para a rede de Kennedy é preciso o ajuste de dois parâmetros.

3.4-) CONCLUSÃO

Neste capítulo, a rede de Hopfield modificada foi utilizada para resolver problemas de otimização linear e quadrática. Os parâmetros internos da rede, correspondentes aos termos de restrições do problema, foram calculados utilizando a técnica do subespaço-válido de soluções. O controlador lógico nebuloso associado à rede, permitiu reduzir o número de iterações da rede sem prejudicar a precisão dos valores finais dos pontos de equilíbrio. Uma análise de complexidade da rede proposta relativa a outras abordagens foi também apresentada. Desta análise, pode-se constatar que a rede proposta é tratável computacionalmente, além de apresentar a vantagem de não possuir nenhum parâmetro intrínseco a ser inicializado, e do qual depende o comportamento final da rede.

Resultados de simulações mostraram que a abordagem proposta apresenta bom desempenho quando comparada com outras arquiteturas de redes neurais. Estes resultados ilustram que a rede desenvolvida pode também ser utilizada como uma alternativa viável em relação às outras abordagens neurais propostas na literatura.

BIBLIOGRAFIA

- [Aho74] A. V. Aho, J. E. Hopcroft and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, Massachusetts, 1974.
- [Baza82] M. S. Bazaraa and J. J. Jarvis. *Linear Programming and Network Flows*. John Wiley & Sons, New York, NY, 1982.
- [Bren81] R. P. Brent and H. T. Kung. "The area-time complexity of binary multiplication". *J. Asso. Comp. Mach.*, vol. 28, pp. 521-534, 1981.
- [Gare79] M. R. Garey and D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.
- [Gima89] C. Gimarc, V. Milutinović and O. Ersoy. "Time complexity of binary modeling and comparison of parallel architectures for Fourier transform oriented algorithms". *Proc. of the Hawaii Int. Conf Syst. Sciences*, HICSS-22, vol. 1, pp. 160-170, 1989.
- [Kenn88] M. P. Kennedy and L. O. Chua. "Neural networks for nonlinear programming". *IEEE Trans. on Circuits and Systems*, vol. 35, pp. 554-562, 1988.

- [Lewi81] H. R. Lewis and C. H. Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [Maa92] C. Y. Maa and M. A. Shamblat. “Linear and quadratic programming neural network analysis”. *IEEE Trans. on Neural Networks*, vol. 3, no. 4, 1992.
- [Ozan86] T. M. Ozan. *Applied Mathematical Programming for Production and Engineering Management*. Prentice-Hall, Englewood Cliffs, NJ, 1986.
- [Papa82] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [Rodr90] A. Rodríguez-Vázquez, R. Domínguez-Castro, A. Rueda, J. L. Huertas and Sánchez-Sinencio. “Nonlinear switched-capacitor ‘neural’ networks for optimization problems”. *IEEE Trans. on Circuits and Systems*, vol. 37, pp. 384-397, 1990.
- [Tank86] D. W. Tank and J. J. Hopfield. “Simple neural optimization networks: an A/D converter, signal decision network, and a linear programming circuit”. *IEEE Trans. on Circuits and Systems*, vol. CAS-33, pp. 533-545, 1986.
- [Zak95] S. H. Zak, V. Upatising and S. Hui. “Solving linear programming problems with neural networks”. *IEEE Trans. on Neural Networks*, vol. 6, no. 1, pp. 94-104, 1995.
- [Zhan92] S. Zhang, X. Zhu and L. H. Zou, “Second-order neural nets for constrained optimization”. *IEEE Trans. on Neural Networks*, vol. 3, no. 6, pp. 1021-1024, 1992.

CAPÍTULO 4

OTIMIZAÇÃO NÃO-LINEAR ATRAVÉS DE REDES NEURAIIS ARTIFICIAIS

4.1-) INTRODUÇÃO

Problemas de otimização não-linear referem-se, geralmente, à otimização de uma função não-linear que pode estar sujeita ou não a um conjunto de restrições (lineares e/ou não-lineares) de igualdade e/ou desigualdade. O termo ‘problema de otimização restrito’ refere-se ao problema de minimizar ou maximizar uma função objetivo na presença destas restrições. Se nenhuma restrição estiver associada à função objetivo, o problema será dito um ‘problema de otimização irrestrito’.

Neste capítulo, a rede neural proposta anteriormente é usada para resolver problemas de otimização restritos e irrestritos que são convexos e contínuos. Inicialmente, mapeiam-se os problemas de otimização restritos (programação não-linear) através da rede de Hopfield modificada. Problemas de otimização irrestritos serão tratados como um caso particular dos problemas de otimização restritos.

A metodologia utilizada neste mapeamento é a mesma proposta no Capítulo 2 e utilizada no Capítulo 3 para tratar problemas de programação linear. Inicialmente, através da utilização de conceitos básicos e definições da teoria de programação não-linear (PNL) e da teoria de estabilidade de Lyapunov, calcula-se as matrizes \mathbf{T}^{val} e \mathbf{s} definindo o subespaço-

válido, e em seguida deduz-se as matrizes \mathbf{T}^{ot} e \mathbf{i}^{ot} , definindo a função objetivo do problema a ser mapeado.

Convém ressaltar que o método proposto é inovador e, diferente da maioria das redes existentes na literatura, para tratar problemas de programação não-linear, não faz uso da técnica de multiplicadores de Lagrange.

Devido a inexistência na literatura de uma arquitetura de rede, passível de comparação com a RHM, nenhuma análise comparativa de resultados é realizada neste capítulo.

A organização deste capítulo está como segue. A Seção 4.2 apresenta alguns conceitos básicos e definições relacionados à teoria de otimização não-linear. O mapeamento de problemas de otimização não-linear através da rede de Hopfield modificada, proposta na Seção 2.4, é apresentado na Seção 4.3. Em seguida, a rede desenvolvida é aplicada na solução de problemas de otimização restritos e irrestritos. Resultados de simulações são também incluídos para mostrar o desempenho da abordagem proposta.

4.2-) CONCEITOS FUNDAMENTAIS DE OTIMIZAÇÃO NÃO-LINEAR

Nesta seção, introduz-se alguns conceitos fundamentais, relacionados à otimização não-linear, que serão utilizados neste capítulo. Estes conceitos auxiliarão na análise das condições necessárias para a convergência da rede em direção aos pontos de equilíbrio.

Dado um conjunto de restrições $\mathbf{h}(\mathbf{x})$, formado por funções contínuas e diferenciáveis, definido por:

$$h_1(\mathbf{x}) = 0$$

$$h_2(\mathbf{x}) = 0$$

...

$$h_p(\mathbf{x}) = 0$$

tem-se da teoria de sistema não-lineares, os seguintes teoremas e definições:

Definição 4.1 [Luen84]: Um ponto regular \mathbf{x}^* satisfazendo as restrições $\mathbf{h}(\mathbf{x}^*)$ é considerado um ponto regular das restrições se os valores dos gradientes:

$$\nabla h_1(\mathbf{x}^*), \nabla h_2(\mathbf{x}^*), \dots, \nabla h_p(\mathbf{x}^*)$$

forem linearmente independentes, onde a matriz gradiente $\nabla \mathbf{h}(\mathbf{x})$ é definida por:

$$\nabla \mathbf{h}(\mathbf{x}) = \begin{bmatrix} \frac{\partial h_1(\mathbf{x})}{\partial x_1} & \frac{\partial h_1(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial h_1(\mathbf{x})}{\partial x_N} \\ \frac{\partial h_2(\mathbf{x})}{\partial x_1} & \frac{\partial h_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial h_2(\mathbf{x})}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_p(\mathbf{x})}{\partial x_1} & \frac{\partial h_p(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial h_p(\mathbf{x})}{\partial x_N} \end{bmatrix} = \begin{bmatrix} \nabla h_1(\mathbf{x})^T \\ \nabla h_2(\mathbf{x})^T \\ \vdots \\ \nabla h_p(\mathbf{x})^T \end{bmatrix}$$

Definição 4.2 [Baza79]: Um conjunto $C \subseteq \mathfrak{R}^N$ é considerado convexo se para todo $\mathbf{x}_1, \mathbf{x}_2 \in C$, o ponto $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in C$, onde $\lambda \in [0, 1]$.

Definição 4.3 [Baza79]: Um ponto \mathbf{x} pertencente a um conjunto convexo C é considerado um ponto extremo de C se não existirem pontos $\mathbf{x}_1, \mathbf{x}_2 \in C$, tal que $\mathbf{x} = \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$, onde $\lambda \in [0, 1]$.

Definição 4.4 [Baza79]: Toda matriz de projeção é simétrica e semidefinida positivamente.

Teorema 4.1 [Luen84]: Num ponto regular \mathbf{x}^* da superfície S definida por $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, o plano tangente M é definido por:

$$M \triangleq \{ \mathbf{x}: \nabla \mathbf{h}(\mathbf{x}^*) \cdot \mathbf{x} = \mathbf{0} \}$$

Teorema 4.2 [Baza79]: Dado que C é um conjunto convexo em \mathfrak{R}^N , a minimização de uma função $f(\mathbf{x})$, sujeita a $\mathbf{x} \in C$, tem uma solução ótima global se $f(\mathbf{x})$ for convexa.

Teorema 4.3 [Baza79]: Seja $f(\mathbf{x})$ uma função duas vezes diferenciável num ponto \mathbf{x}^* . Se \mathbf{x}^* for um ponto de mínimo, então $\nabla f(\mathbf{x}^*) = \mathbf{0}$, e a matriz Hessiana $\mathbf{H}(\mathbf{x}^*)$, cujos elementos são as derivadas parciais de segunda ordem de $f(\mathbf{x}^*)$, é semidefinida positivamente.

As definições e teoremas acima também se aplicam ao conjunto de restrições $\mathbf{g}(\mathbf{x})$, formado por funções contínuas e diferenciáveis, definido como:

$$\begin{aligned} g_1(\mathbf{x}) &\leq 0 \\ g_2(\mathbf{x}) &\leq 0 \\ &\dots \\ g_d(\mathbf{x}) &\leq 0 \end{aligned}$$

Na próxima seção, utilizam-se estes conceitos na derivação dos parâmetros internos da rede de Hopfield modificada, a ser aplicada na solução de problemas de otimização não-linear.

4.3-) PROBLEMAS DE OTIMIZAÇÃO NÃO-LINEAR

Os algoritmos convencionais que resolvem o problema de otimização não-linear de forma eficiente, em geral, utilizam algum procedimento de busca para encontrar uma sequência de soluções, que conduza à solução ótima segundo algum critério. Uma descrição detalhada destes métodos é apresentado em [Luen84, Baza79].

Recentemente, alguns métodos utilizando RNA, na solução de problemas de otimização não-linear, têm sido desenvolvidos. Os principais métodos neurais usados na solução de problemas de otimização restritos são baseados na técnica dos multiplicadores de Lagrange [Baza79]. Entre estes métodos, destacam-se: a rede neural de segunda ordem desenvolvida em [Zhan92], o método do multiplicador diferencial básico (BDMM) proposto em [Plat87], o método de otimização em duas fases apresentado em [Maa92], e o método do subgradiente desenvolvido em [Rome93]. Na próxima subseção, um novo método é desenvolvido, através da RHM proposta na Seção 2.4, para solucionar problemas de otimização não-linear.

4.3.1-) Mapeamento de Problemas de Otimização Restritos Através da Rede de Hopfield Modificada

Problemas de otimização restritos, com p-restrições de igualdade e q-restrições de desigualdade, podem ser descritos pelas seguintes equações:

$$\text{Minimizar } E^{\text{ot}}(\mathbf{v}) = f(\mathbf{v}) \tag{4.1}$$

$$\text{sujeito a: } E^{\text{conf}}(\mathbf{v}) \begin{cases} h_i(\mathbf{v}) = 0, & i \in \{1..p\} \\ g_j(\mathbf{v}) \leq 0, & j \in \{1..q\} \end{cases} \tag{4.2}$$

$$\mathbf{z}^{\text{min}} \leq \mathbf{v} \leq \mathbf{z}^{\text{max}} \tag{4.4}$$

onde: $\mathbf{v}, \mathbf{z}^{\text{min}}, \mathbf{z}^{\text{max}} \in \mathfrak{R}^N$; $\mathbf{g}(\mathbf{v}) \leq \mathbf{0}$ e $\mathbf{h}(\mathbf{v}) = \mathbf{0}$ definem um conjunto convexo em \mathfrak{R}^N , e as funções $f(\mathbf{v})$, $h_i(\mathbf{v})$ e $g_j(\mathbf{v})$ são funções contínuas e diferenciáveis. As equações analíticas para os parâmetros internos da rede, definidos por $\mathbf{T}^{\text{val}}, \mathbf{s}, \mathbf{T}^{\text{ot}}$ e \mathbf{i}^{ot} , são explicitadas a seguir.

Dedução das Equações Para T^{val} e s

Considera-se inicialmente, o caso em que existam apenas restrições de igualdade na forma descrita por (4.2). Para sistemas não-lineares, diferentemente do procedimento utilizado para a derivação de T^{val} e s no caso de sistemas lineares, é difícil estabelecer alguma suposição sobre uma solução admissível inicial, que resulte na derivação de um subespaço-válido de soluções. Entretanto, a partir da teoria de estabilidade de Lyapunov [Vidy93a], é possível fazer análises sobre um sistema não-linear, baseando-se no comportamento de um sistema linear que aproxime o sistema original em torno de um determinado ponto de equilíbrio.

A evolução de um sistema dinâmico não-linear (autônomo) pode ser representada, geralmente, por um sistema de equações diferenciais de primeira ordem com a seguinte forma [Rose70]:

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}(t)}{dt} = \mathbf{F}(\mathbf{x}(t)) \quad (4.5)$$

onde a função $\mathbf{F}(\mathbf{x}(t))$ representa um conjunto de equações (em geral) não-lineares, na forma dada pela equação (4.2). Assim, considerando um sistema dinâmico descrito pela equação (4.5), um vetor \mathbf{x}^e é considerado um estado de equilíbrio do sistema se for satisfeita a seguinte equação [Fang96, Vidy93a]:

$$\mathbf{F}(\mathbf{x}^e) = \mathbf{0} \quad (4.6)$$

onde $\mathbf{F}(\cdot)$ é continuamente diferenciável.

Desenvolvendo a função $\mathbf{F}(\mathbf{x})$ a partir dos dois primeiros termos da série de Taylor, o valor de $\mathbf{F}(\mathbf{x})$ avaliado na vizinhança de \mathbf{x}^e é dado por:

$$\mathbf{F}(\mathbf{x}) \cong \mathbf{F}(\mathbf{x}^e) + \mathbf{A} \cdot (\mathbf{x} - \mathbf{x}^e) \quad (4.7)$$

onde a matriz \mathbf{A} é definida como o Jacobiano de $\mathbf{F}(\mathbf{x})$ em relação a \mathbf{x} , ou seja, $\mathbf{A} = \nabla \mathbf{F}(\mathbf{x})$.

Assim, analisando-se o valor de $\mathbf{F}(\mathbf{x})$ na vizinhança do estado $\mathbf{x}^e = \mathbf{0}$, obtém-se a condição de equilíbrio dada em (4.6), ou seja:

$$\lim_{\mathbf{x} \rightarrow \mathbf{x}^e} \frac{\|\mathbf{F}(\mathbf{x})\|}{\|\mathbf{x}\|} = \mathbf{0} \quad (4.8)$$

onde o símbolo $\|\cdot\|$ denota a norma Euclidiana. Logo, levando em consideração esta condição, a equação (4.7), torna-se:

$$\mathbf{F}(\mathbf{x}) \cong \mathbf{A} \cdot \mathbf{x} = \nabla \mathbf{F}(\mathbf{x}) \cdot \mathbf{x} \quad (4.9)$$

É importante notar que a equação (4.9) é válida, sem perda de generalidade, para qualquer ponto de equilíbrio $\mathbf{x}^e \in \mathcal{R}^N$. Para um ponto de equilíbrio onde $\mathbf{x}^e \neq \mathbf{0}$, pode-se transladar os eixos coordenados a fim de que a origem do novo sistema seja coincidente com \mathbf{x}^e [Fang96]. Baseado nestes conceitos, deduz-se portanto, que uma solução para o sistema de equações não-lineares, dada por (4.2), é o próprio vetor $\mathbf{v}^e = \mathbf{0}$, que será introduzido na equação do subespaço-válido através do vetor \mathbf{s} .

Por outro lado, a matriz projeção \mathbf{T}^{val} da equação do subespaço-válido é obtida através da projeção de \mathbf{v} , obtido a partir da minimização de $E^{\text{ot}}(\mathbf{v}) = f(\mathbf{v})$, para o subespaço tangente à superfície delimitada pelas restrições dadas por (4.2) [Baza79]. Assim, uma equação para \mathbf{T}^{val} pode ser definida, a partir da equação linearizada (4.9), por:

$$\mathbf{T}^{\text{val}} = \mathbf{I} - \nabla \mathbf{h}(\mathbf{v})^T \cdot (\nabla \mathbf{h}(\mathbf{v}) \cdot \nabla \mathbf{h}(\mathbf{v})^T)^{-1} \cdot \nabla \mathbf{h}(\mathbf{v}) \quad (4.10)$$

Substituindo-se a matriz \mathbf{T}^{val} na equação do subespaço-válido, obtém-se:

$$\mathbf{v} = [\mathbf{I} - \nabla \mathbf{h}(\mathbf{v})^T \cdot (\nabla \mathbf{h}(\mathbf{v}) \cdot \nabla \mathbf{h}(\mathbf{v})^T)^{-1} \cdot \nabla \mathbf{h}(\mathbf{v})] \cdot \mathbf{v} + \mathbf{s} \quad (4.11)$$

Por fim, as condições deduzidas a partir de (4.8) e (4.9) devem ser introduzidas na equação (4.11) para garantir a estabilidade do sistema não-linear, e conseqüentemente, forçar a convergência para os pontos de equilíbrio que representam uma solução para o sistema. Para tanto, desenvolvendo-se a equação (4.11), tem-se:

$$\mathbf{v} = \mathbf{I} \cdot \mathbf{v} - \nabla \mathbf{h}(\mathbf{v})^T \cdot (\nabla \mathbf{h}(\mathbf{v}) \cdot \nabla \mathbf{h}(\mathbf{v})^T)^{-1} \cdot \nabla \mathbf{h}(\mathbf{v}) \cdot \mathbf{v} + \mathbf{s} \quad (4.12)$$

Utilizando-se o resultado obtido em (4.9), conclui-se que quando \mathbf{v} ($\|\mathbf{s}\| \rightarrow \mathbf{0}$) tende ao equilíbrio, a equação (4.12) torna-se:

$$\mathbf{v} = \mathbf{v} - \nabla \mathbf{h}(\mathbf{v})^T \cdot (\nabla \mathbf{h}(\mathbf{v}) \cdot \nabla \mathbf{h}(\mathbf{v})^T)^{-1} \cdot \mathbf{h}(\mathbf{v}) \quad (4.13)$$

Logo, a equação (4.13) sintetiza a equação do subespaço-válido para sistemas de equações não-lineares. Neste caso, a equação original do subespaço-válido em (2.9), representando a minimização indireta de E^{conf} , deve ser substituída pela equação (4.13). A

aplicação sucessiva do passo (I) seguido do passo (II) da Figura (2.2), faz com que \mathbf{v} seja uma solução que satisfaça todas as restrições impostas pelo problema de otimização não-linear restrito.

A análise realizada para obtenção da equação (4.13) pode também ser feita com o auxílio do Teorema 4.1, que mostra que num ponto regular de $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$, a equação do plano tangente é definida por $\nabla \mathbf{h}(\mathbf{x}^*) \cdot \mathbf{x} = \mathbf{0}$. Assim, levando em conta todas estas considerações, a equação (4.13) pode ser visualizada como o confinamento (indireto) de \mathbf{v} no subespaço-válido tangente à superfície delimitada por todas as restrições ($\mathbf{h}(\mathbf{v}) = \mathbf{0}$) impostas pelo problema de otimização restrito. É importante ressaltar que as propriedades do subespaço-válido são verificadas também para o caso não-linear, ou seja, $(\mathbf{T}^{\text{val}})^T \mathbf{T}^{\text{val}} = \mathbf{T}^{\text{val}}$ e $(\mathbf{T}^{\text{val}})^T \mathbf{s} = \mathbf{0}$.

Como no caso linear, a metodologia utilizada para as restrições de igualdade não-linear pode também ser estendida para as restrições de desigualdade não-linear dada por (4.3). Logo, uma restrição de desigualdade típica dada por $g_i(\mathbf{v}) \leq 0$, torna-se:

$$g_i(\mathbf{v}) + \sum_{j=1}^q q_j \cdot w_j = 0 \tag{4.14}$$

onde w_j são variáveis auxiliares (tratadas como variáveis do vetor \mathbf{v}), e q_j são constantes definidas pela função impulso de Kronecher (3.11).

Dedução das Equações Para \mathbf{T}^{ot} e \mathbf{i}^{ot}

Os parâmetros \mathbf{T}^{ot} e \mathbf{i}^{ot} do termo de energia E^{ot} , dado por (2.6), são definidos com o objetivo de que a solução ótima corresponda à minimização de E^{ot} . Este procedimento pode ser feito, dado que $f(\mathbf{v})$ é uma função convexa, com a utilização do Teorema 4.3. Assim, a condição para que \mathbf{v} seja um ponto de mínimo da função $E^{\text{ot}}(\mathbf{v}) = f(\mathbf{v})$ é dada por:

$$-\nabla E^{\text{ot}}(\mathbf{v}) = \mathbf{T}^{\text{ot}} \cdot \mathbf{v} + \mathbf{i}^{\text{ot}} = \mathbf{0} \tag{4.15}$$

Comparando a equação (4.15) com os resultados advindos do Teorema 4.3, obtêm-se os parâmetros \mathbf{T}^{ot} e \mathbf{i}^{ot} , definidos por:

$$\mathbf{i}^{\text{ot}} = - \left[\frac{\partial f(\mathbf{v})}{\partial v_1}, \frac{\partial f(\mathbf{v})}{\partial v_2}, \dots, \frac{\partial f(\mathbf{v})}{\partial v_N} \right] \tag{4.16}$$

$$\mathbf{T}^{\text{ot}} = \mathbf{0} \tag{4.17}$$

A análise do comportamento dinâmico da rede de Hopfield modificada, realizada na Seção 2.5, pode ser aplicada, sem perda de generalidade, ao mapeamento proposto neste capítulo. Como visto anteriormente, para sistemas não-lineares, tem-se que numa pequena vizinhança em torno dos pontos de equilíbrio, o sistema não-linear se comporta como linear. Neste caso, assim como no caso linear, verifica-se que a matriz \mathbf{T}^{val} torna-se constante em torno do ponto de equilíbrio $\mathbf{x}^e = \mathbf{0}$. Esta análise, onde as matrizes de conexões da rede (\mathbf{T}^{val} e \mathbf{T}^{ot}) são simétricas e constantes em torno dos pontos de equilíbrio é também realizada em [Vidy93b]. Assim, a rede de Hopfield modificada, cujo subespaço-válido é calculado por (4.10) e (4.11), e com função objetivo mapeada através de (4.16) e (4.17), quando aplicada a um problema de otimização não-linear da forma (4.1)-(4.4) sempre converge para uma solução válida. Do mesmo modo, os problemas de otimização linear e quadrática (Capítulo 3) podem também ser resolvidos, com alguma perda de simplicidade, através da rede desenvolvida neste capítulo, confirmando assim a sua generalidade. Vale notar que para o caso de otimização restrita, a rede implementa de forma indireta o método gradiente projetado [Baza79, Luen84]. Descreve-se a seguir, os resultados de simulação obtidos com a rede proposta.

Resultados de Simulação

A rede de Hopfield modificada foi aplicada aos seguintes problemas restritos de otimização não-linear:

Problema 1-) Seja o problema de otimização restrito definido por:

$$\text{Min } f(\mathbf{v}) = e^{v_1} + v_1^2 + 4v_1 + 2v_2^2 - 6v_2 + 2v_3$$

$$\text{sujeito a: } v_1^2 + e^{v_2} + 6v_3 \leq 15 \quad (\text{R1})$$

$$v_1^4 - v_2 + 5v_3 \leq 25 \quad (\text{R2})$$

$$v_1^3 + v_2^2 - v_3 \leq 10 \quad (\text{R3})$$

$$0 \leq v_1 \leq 4 \quad (\text{R4})$$

$$0 \leq v_2 \leq 2 \quad (\text{R5})$$

$$v_3 \geq 0 \quad (\text{R6})$$

Neste problema, com três restrições de desigualdade e com variáveis limitadas (canalizadas), o vetor solução (ponto de equilíbrio) obtido pela rede é dado por $\mathbf{v} = [0.0022 \ 1.5008 \ 0.0000]^T$, com função objetivo $E^{\text{ot}}(\mathbf{v}) = -3.555$. Estes resultados são bem próximos ao valor da solução ótima fornecida por $\mathbf{v}^* = [0.0000 \ 1.5000 \ 0.0000]^T$ e com $E^{\text{ot}}(\mathbf{v}^*) = -3.500$. As restrições R4, R5 e R6, que determinam os limites inferiores e

superiores para as variáveis, são mapeadas diretamente pela função de ativação rampa-simétrica, definida em (2.10). Os valores iniciais atribuídos ao vetor v , foram gerados aleatoriamente entre zero e um.

A Figura 4.1 ilustra a evolução dos valores de v_1 , v_2 e v_3 obtidos pela RHM, em relação ao número de iterações. O esboço do comportamento da função objetivo do problema é apresentado na Figura 4.2.

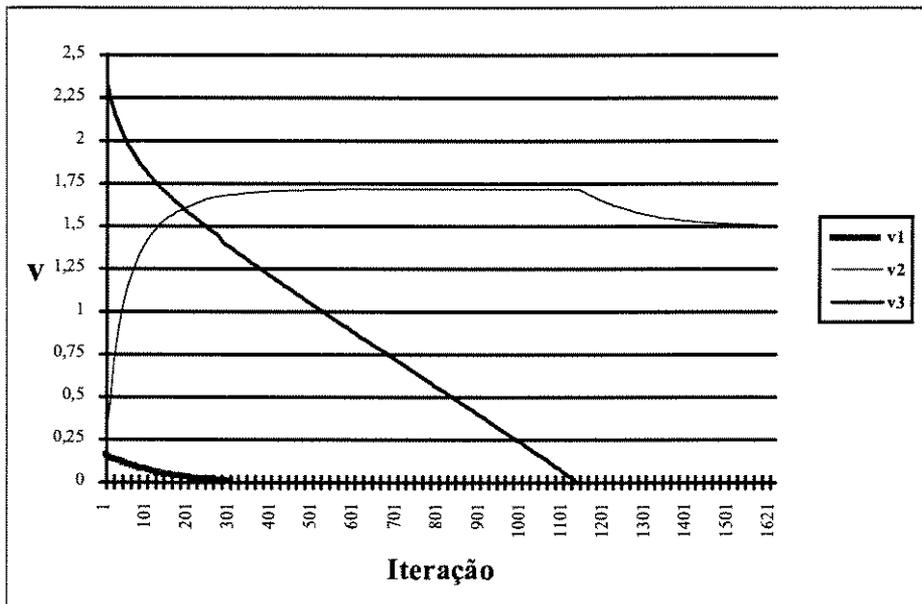


Figura 4.1. Evolução do Vetor de Saída da Rede (Prob. Restrito 1)

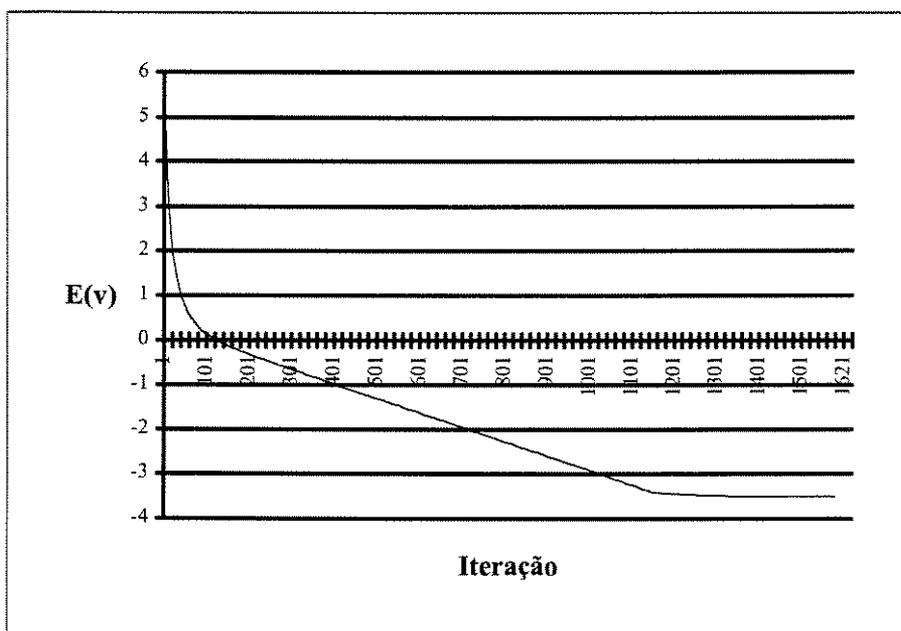


Figura 4.2. Comportamento da Função Objetivo (Prob. Restrito 1)

Problema 2-) Seja o problema dado por:

$$\text{Min } f(\mathbf{v}) = v_1^3 + 2v_2^2 \cdot v_3 + 2v_3$$

$$\text{sujeito a: } v_1^2 + v_2 + v_3^2 = 4$$

$$v_1^2 - v_2 + 2v_3 \leq 2$$

$$v_1, v_2, v_3 \geq 0$$

Para este problema, com uma restrição de igualdade e uma restrição de desigualdade, a Figura 4.3 mostra a evolução dos valores de v_1 , v_2 e v_3 em relação ao número de iterações. O vetor solução obtido, após a convergência da rede, é dado por $\mathbf{v} = [0.000 \ 3.997 \ 0.000]^T$, com $E^{ot}(\mathbf{v}) = 0.0001$. A solução ótima para o problema é fornecida por $\mathbf{v}^* = [0.000 \ 4.000 \ 0.000]^T$, com $E^{ot}(\mathbf{v}^*) = 0$.

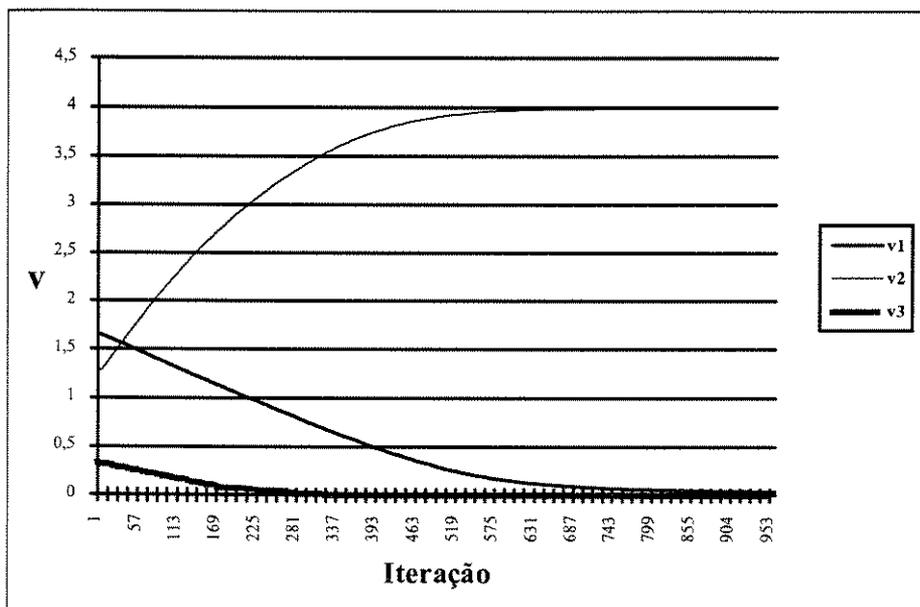


Figura 4.3. Evolução do Vetor de Saída da Rede (Prob. Restrito 2)

A Figura 4.4 ilustra o comportamento da função objetivo do problema, em função do número de iterações. Os valores iniciais atribuídos ao vetor \mathbf{v} , foram gerados aleatoriamente entre zero e um.

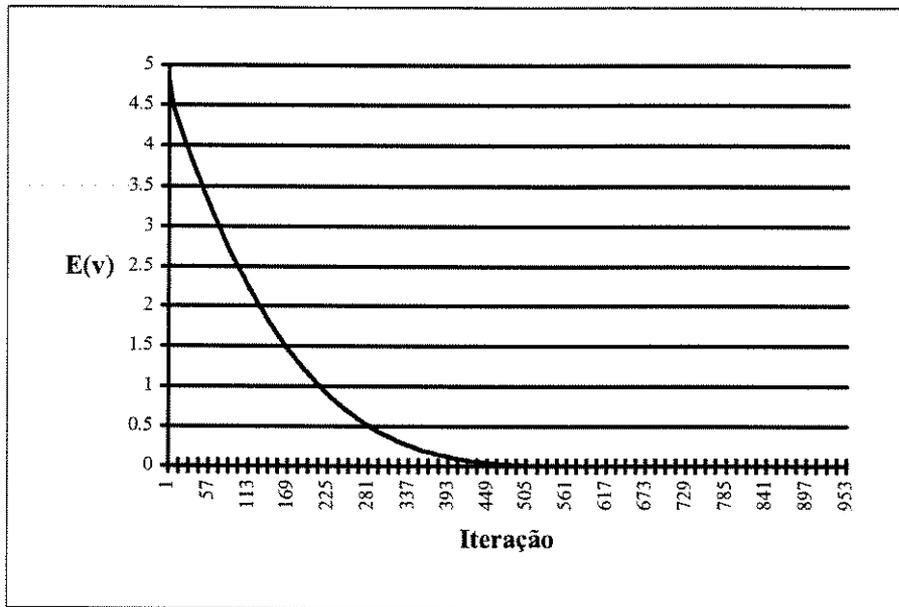


Figura 4.4. Comportamento da Função Objetivo (Prob. Restrito 2)

Problema 3-) Dado o problema definido por [Kenn88]:

$$\begin{aligned} \text{Min } f(\mathbf{v}) &= 0.4v_2 + v_1^2 + v_2^2 - v_1v_2 + \frac{1}{30}v_1^3 \\ \text{sujeito a: } &v_1 + 0.5v_2 \geq 0.4 \\ &0.5v_1 + v_2 \geq 0.5 \\ &v_1, v_2 \geq 0 \end{aligned}$$

Neste exemplo, com duas restrições de desigualdade, confirma-se a generalidade da rede proposta, quando todas as restrições associadas ao problema são do tipo linear. A Figura 4.5 apresenta a evolução dos valores dos elementos do vetor de saída da rede em relação ao número de iterações. Após a convergência da rede, o vetor solução obtido é dado por $\mathbf{v} = [0.3398 \ 0.3301]^T$, com $E^{ot}(\mathbf{v}) = 0.2456$. Estes resultados são bem próximos aos valores da solução exata fornecida por $\mathbf{v}^* = [0.3395 \ 0.3302]^T$, com função objetivo $E^{ot}(\mathbf{v}^*) = 0.2455$.

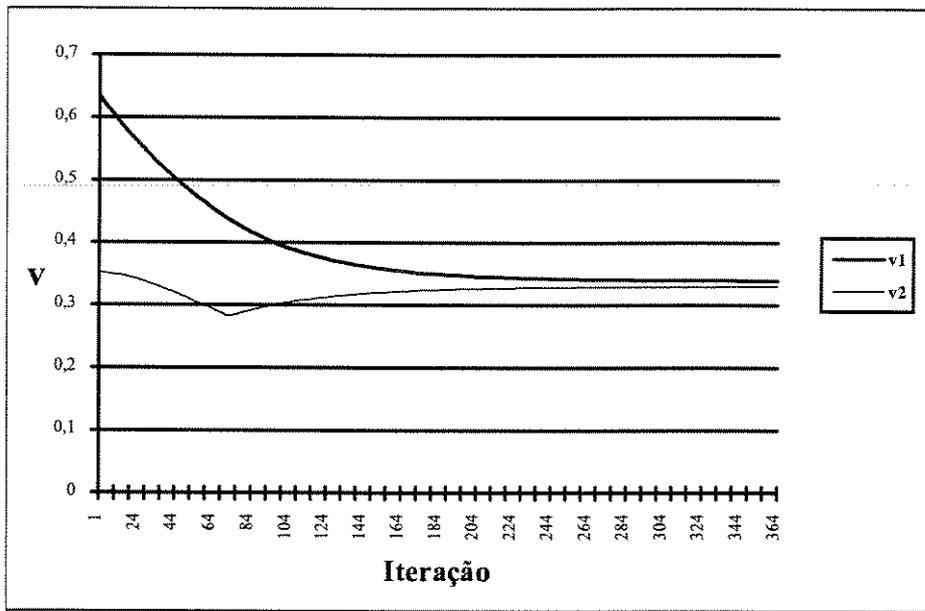


Figura 4.5. Evolução do Vetor de Saída da Rede (Prob. Restrito 3)

A Figura 4.6 apresenta o comportamento da função objetivo, em função do número de iterações. A rede foi inicializada com valores aleatórios, uniformemente distribuídos, entre zero e um.

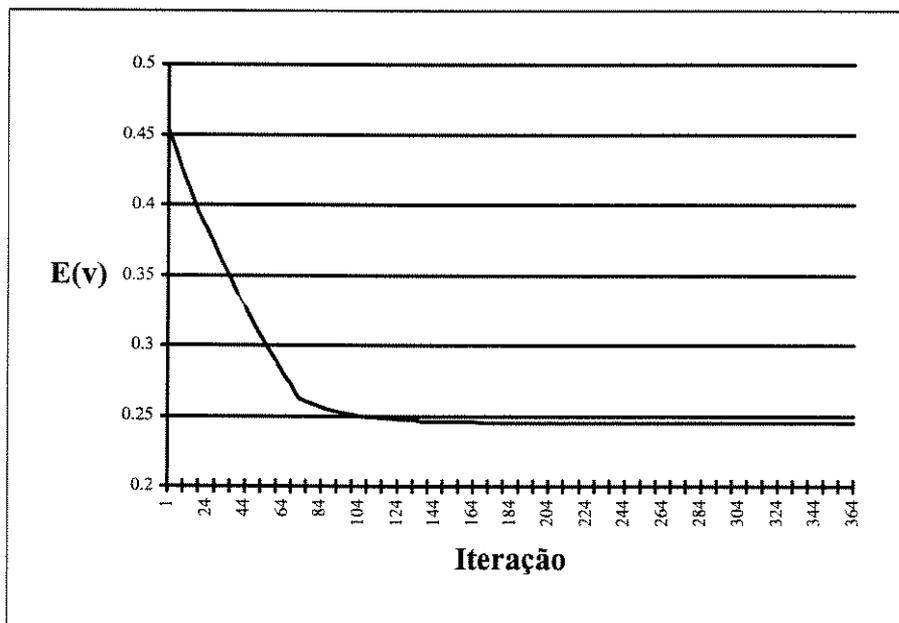


Figura 4.6. Comportamento da Função Objetivo (Prob. Restrito 3)

A seguir descreve-se como problemas de otimização irrestritos são mapeados através da rede proposta neste capítulo.

4.3.2-) Mapeamento de Problemas de Otimização Irrestritos Através da Rede de Hopfield Modificada

Problemas de otimização irrestritos podem ser descritos da seguinte forma:

$$\begin{aligned} \text{Minimizar } E^{ot}(\mathbf{v}) &= f(\mathbf{v}) \\ \mathbf{v} &\in \mathcal{R}^N \end{aligned} \quad (4.18)$$

Entre os métodos utilizados para resolver problemas de otimização irrestritos, destacam-se ‘os métodos do tipo gradiente’ que usam informações sobre a função a ser otimizada e sua respectiva derivada [Baza79]. Entre os métodos gradiente encontram-se ‘o método do gradiente simples’, ‘o método do gradiente conjugado’ e ‘o método quase-Newton’.

Por outro lado, existem os métodos que usam, além das informações sobre a função a ser otimizada, as suas derivadas de primeira e segunda ordem. Entre estes métodos destaca-se ‘o método de Newton’. Em geral, esses métodos são menos utilizados que os métodos que usam apenas a primeira derivada, visto que eles requerem mais informações e conseqüentemente mais tempo de cálculo por iteração. Entretanto, eles geralmente são mais rápidos, especialmente, quando a solução inicial está próxima à solução ótima [Baza79].

Dedução das Equações Para \mathbf{T}^{val} , \mathbf{s} , \mathbf{T}^{ot} e \mathbf{i}^{ot}

Diferentemente dos problemas de otimização restritos, os problemas de otimização irrestritos não requerem qualquer restrição sobre as variáveis da função a ser otimizada. Assim, utilizando a matriz projeção dada em (4.10), tem-se que o valor de $\nabla \mathbf{h}(\mathbf{v})$ é nulo para os problemas de otimização irrestritos. Logo, a matriz projeção \mathbf{T}^{val} do subespaço-válido torna-se:

$$\mathbf{T}^{val} = \mathbf{I} \quad (4.19)$$

Substituindo a equação (4.19) na expressão do subespaço-válido, obtém-se:

$$\begin{aligned} \mathbf{v} &= \mathbf{I} \cdot \mathbf{v} + \mathbf{s} \\ \mathbf{s} &= \mathbf{0} \end{aligned} \quad (4.20)$$

onde as propriedades do subespaço-válido são mantidas, ou seja, $(\mathbf{T}^{val} \cdot \mathbf{T}^{val} = \mathbf{T}^{val})$ e $(\mathbf{T}^{val} \cdot \mathbf{s} = \mathbf{0})$.

Por conseguinte, as equações de T^{ot} e i^{ot} são idênticas àquelas utilizadas para os problemas de otimização restritos, visto que para problemas de otimização irrestritos, se deseja otimizar uma função objetivo sem nenhuma restrição. Vale notar que para o caso de otimização irrestrita, a rede implementa indiretamente o método do gradiente simples. Em seguida, utiliza-se a rede proposta na solução de dois problemas de otimização irrestritos.

Problema 1-) Seja o problema de otimização irrestrito definido por:

$$\text{Min } f(\mathbf{v}) = (v_1 - 0.6)^4 + (v_1 + 2v_2)^2$$

Neste problema, a solução obtida pela rede, após a convergência, é dada por $\mathbf{v} = [0.6007 \ 0.3006]^T$, com $E^{ot}(\mathbf{v}) = 2.31 \times 10^{-7}$. Estes valores são bem próximos aos valores ótimos para o problema, que são fornecidos por $\mathbf{v}^* = [0.6000 \ 0.3000]^T$, com função $E^{ot}(\mathbf{v}^*) = 0$.

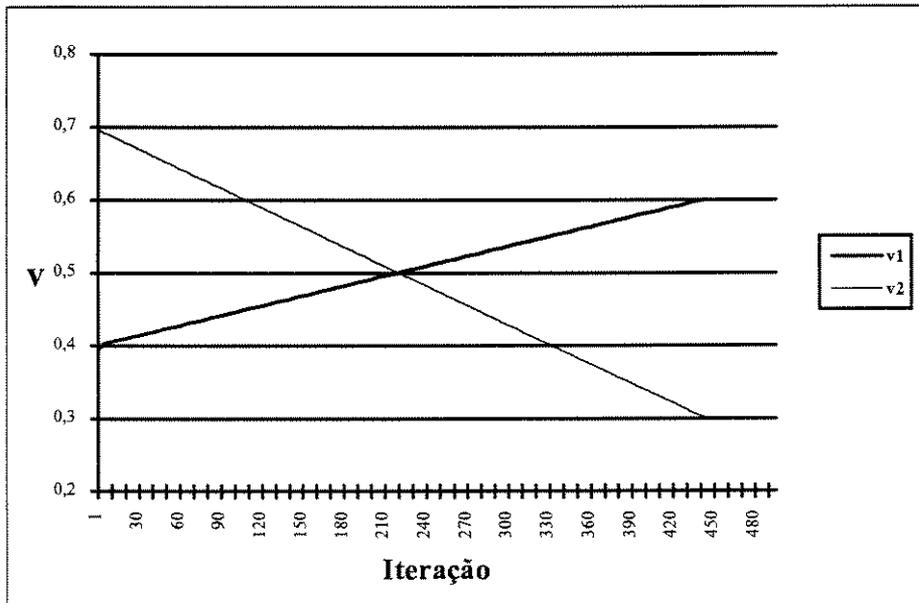


Figura 4.7. Evolução do Vetor de Saída da Rede (Prob. Irrestrito 1)

A Figura 4.7 apresenta a evolução dos valores dos elementos de \mathbf{v} , em relação ao número de iterações. A Figura 4.8 ilustra o comportamento da função objetivo do problema, em função do número de iterações.

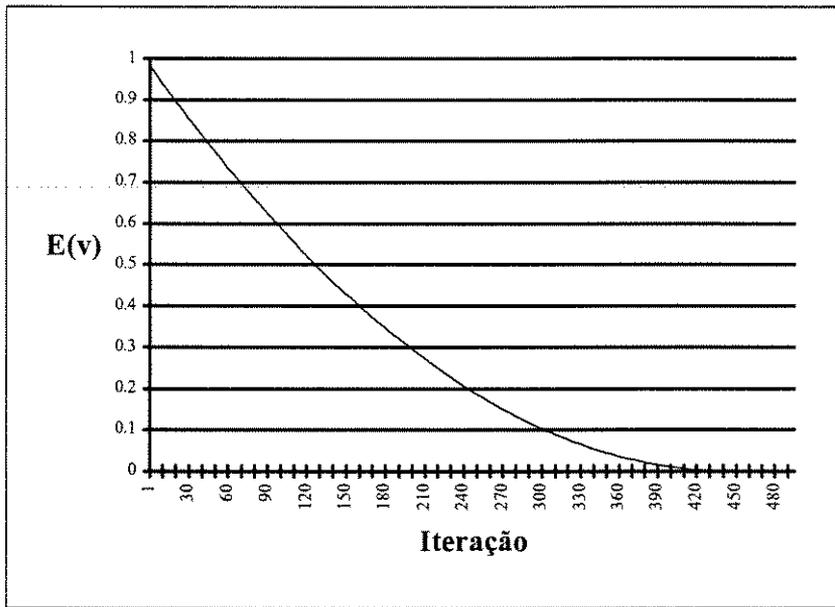


Figura 4.8. Comportamento da Função Objetiva (Prob. Irrestrito 1)

Problema 2-) Considera-se o problema definido por:

$$\text{Min } f(\mathbf{v}) = (v_1 - 0.1)^2 + (v_2 - 0.3)^2 + (v_3 - 0.5)^2$$

Para este problema de otimização irrestrita, a Figura 4.9 ilustra a evolução dos valores de v_1 , v_2 e v_3 , em relação ao número de iterações. O vetor solução obtido, após a convergência da rede, é dado por $\mathbf{v} = [0.1002 \ 0.2998 \ 0.5001]^T$, com função objetivo $E^{ot}(\mathbf{v}) = 1.23 \times 10^{-7}$. A solução ótima para o problema é fornecida por $\mathbf{v}^* = [0.1000 \ 0.3000 \ 0.5000]^T$, com $E^{ot}(\mathbf{v}^*) = 0$.

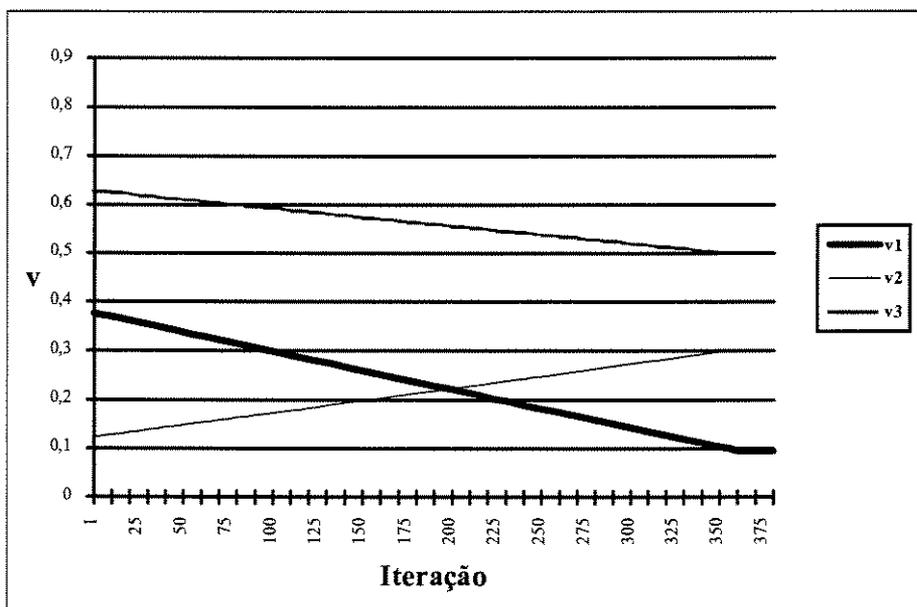


Figura 4.9. Evolução do Vetor de Saída da Rede (Prob. Irrestrito 2)

O esboço do comportamento da função objetivo do problema é apresentado na Figura 4.10. A rede foi simulada, para todos os problemas, considerando diversos valores iniciais atribuídos ao vetor de saída v . Para todas as simulações, a rede sempre convergiu para as mesmas soluções.

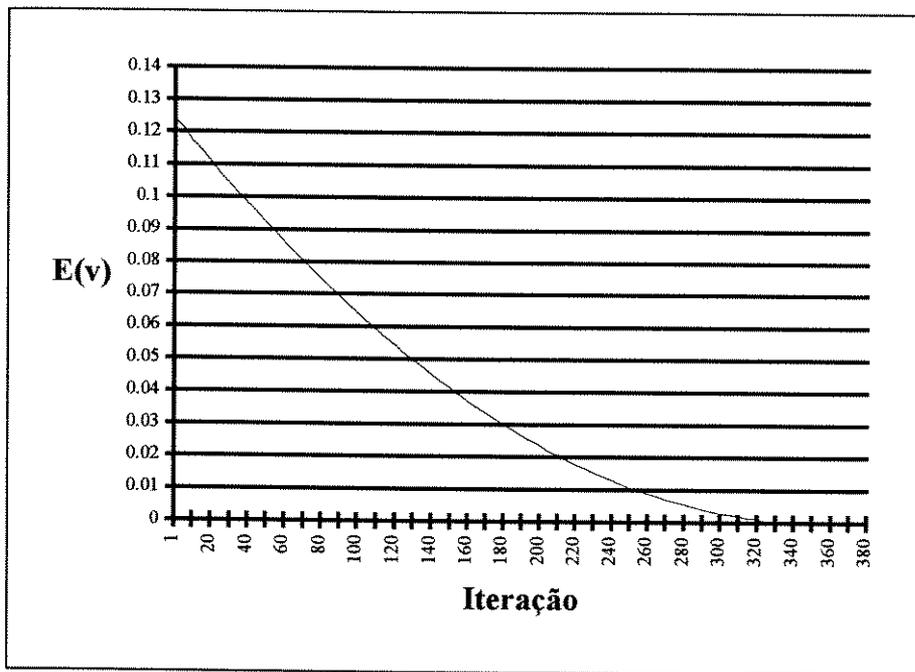


Figura 4.10. Comportamento da Função Objetivo (Prob. Irrestrito 2)

4.4-) CONCLUSÃO

Neste capítulo, propôs-se um novo método, utilizando a rede de Hopfield modificada, para resolver problemas de otimização não-linear. Aplicou-se a abordagem desenvolvida na solução de problemas de otimização restritos e irrestritos. Os problemas de otimização irrestritos são tratados como um caso particular de problemas restritos. Neste caso, a rede de Hopfield modificada implementa o método do gradiente simples. Os parâmetros internos da rede foram obtidos utilizando conceitos relativos às teorias de otimização não-linear e de estabilidade de Lyapunov. Todas as restrições (igualdades e desigualdades) associadas aos problemas de otimização restritos são mapeadas através do subespaço-válido de soluções.

Os resultados de simulações mostraram que a rede proposta fornece bons resultados, tanto para problemas de otimização restritos como para problemas irrestritos. Destes resultados, considera-se que a rede desenvolvida pode também ser utilizada como uma alternativa viável às arquiteturas de redes neurais que utilizam multiplicadores de Lagrange.

BIBLIOGRAFIA

- [Baza79] M. S. Bazaraa and C. M. Shetty. *Nonlinear Programming - Theory and Algorithms*. John Wiley & Sons, New York, NY, 1979.
- [Fang96] Y. Fang and T. G. Kincaid. "Stability analysis of dynamical neural networks". *IEEE Trans. on Neural Networks*, vol. 7, no. 4, pp. 996-1005, 1996.
- [Kenn88] M. P. Kennedy and L. O. Chua. "Neural network for nonlinear programming". *IEEE Trans. on Circuits and Systems*, vol. 35, no. 5, pp. 554-562, 1988.
- [Luen84] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Reading, Massachusetts, 1984.
- [Maa92] C. -Y. Maa and M. A. Shanblatt. "A two phase optimization neural network". *IEEE Trans. on Neural Networks*, vol. 3, no. 6, 1992.
- [Plat87] J. Platt and Al Barr. "Constrained differential optimization". *Proc. of the Neural Information Processing Systems Conference*, 1987.
- [Rome93] R. A. F. Romero. *Otimização de Sistemas Através de Redes Neurais Artificiais*. Tese de Doutorado, DCA/FEE/UNICAMP, Campinas-SP, Julho de 1993.
- [Rose70] H. H. Rosenbrook and C. Storey. *Mathematics of Dynamical Systems*. Thomas Nelson & Sons, London, 1970.
- [Vidy93a] M. Vidyasagar. *Nonlinear Systems Analysis (Second Edition)*. Prentice Hall, Englewood Cliffs, NJ, 1993.
- [Vidy93b] M. Vidyasagar. "Location and stability of the high-gain equilibria of nonlinear neural networks". *IEEE Trans. on Neural Networks*, vol. 4, no. 4, 1993.

[Zhan92] S. Zhang, X. Zhu and L. -H. Zou. “Second-order neural nets for constrained optimization”. *IEEE Trans. on Neural Networks*, vol. 3, no. 6, pp. 1021-1024, 1992.

CAPÍTULO 5

OTIMIZAÇÃO COMBINATORIAL E PROGRAMAÇÃO DINÂMICA POR REDES NEURAIS ARTIFICIAIS

5.1-) INTRODUÇÃO

A proposta deste capítulo é utilizar a rede de Hopfield modificada na resolução de problemas de otimização combinatorial e programação dinâmica através da metodologia descrita no Capítulo 2.

A utilização de Redes Neurais Artificiais em problemas de programação combinatorial, além de oferecer uma nova e poderosa heurística para resolver estes problemas, fornece a vantagem de utilizar arquiteturas de processamento inerentemente adaptativas e paralelas, com potencial para implementação rápida em hardware. As origens das pesquisas nesta área podem ser atribuídas aos trabalhos de J. J. Hopfield [Hopf85], e aos trabalhos realizados por vários pesquisadores sobre a mecânica estatística do sistema Vidro de Spin (Spin-Glass Systems). Neste capítulo, utiliza-se a rede de Hopfield modificada na solução do problema do caixeiro viajante, das N-rainhas e do matching-bipartido.

O problema do caixeiro viajante foi inicialmente solucionado por Hopfield e Tank [Hopf85] através da rede de Hopfield convencional apresentada no Capítulo 2. No entanto, como discutido anteriormente, esta rede para convergir corretamente aos pontos de equilíbrio,

depende de uma especificação adequada de seus parâmetros de ponderação. Ao contrário, a rede de Hopfield modificada, não necessita da definição de nenhum parâmetro de ponderação.

O problema de otimização combinatorial conhecido como o ‘problema das N-rainhas’, já foi solucionado através de uma rede neural [Mand92]. Como no caso do problema do caixeiro viajante, a convergência da rede depende de uma especificação adequada de seus parâmetros de ponderação, caso contrário, a solução obtida pode ser infactível. Um outro problema de otimização combinatorial, tratado também neste capítulo, é o ‘problema do matching-bipartido’.

Em relação aos problemas de programação dinâmica, um novo método para mapear o ‘problema do caminho mínimo’ é desenvolvido com a utilização da RHM. As soluções obtidas com este novo método, para diferentes situações deste problema, são comparadas àquelas obtidas com a rede neural proposta em [Chiu91]

A Seção 5.2 apresenta a notação utilizada nos problemas de otimização combinatorial e de programação dinâmica. A Seção 5.3 descreve o mapeamento dos problemas de programação combinatorial através da rede de Hopfield modificada. Como exemplo, o problema do caixeiro viajante, o problema das N-rainhas e o problema do matching-bipartido são solucionados, e os resultados de simulações para estes exemplos são utilizados na comparação com outras abordagens. Na Seção 5.4, mostra-se como o problema do caminho mínimo que é um problema clássico de programação dinâmica, pode ser resolvido com a rede proposta neste trabalho. Uma análise de complexidade da RHM, aplicada na solução dos problemas apresentados neste capítulo, é realizada no Apêndice E.

5.2-) NOTAÇÕES, DEFINIÇÕES E PROPRIEDADES

Uma característica fundamental dos problemas, que são tratados neste capítulo, é a existência de um conjunto finito de n -soluções, sendo possível representar cada solução por um inteiro ou até mesmo um conjunto de inteiros. Além disso, os vetores que representam cada uma destas soluções possuem tipicamente elementos com valores zero ou um. Assim, dado um vetor $\mathbf{p} \in \mathcal{R}^n$ representando cada uma destas soluções, cujos elementos são inteiros no intervalo $\{1, \dots, m\}$, tem-se:

$$p_i \in \{1, \dots, m\} \quad \text{onde } i \in \{1..n\} \quad (5.1)$$

Como um exemplo específico, considera-se o bem conhecido problema do caixeiro viajante, onde o objetivo é encontrar o menor caminho (percurso) entre várias cidades que devem ser visitadas uma única vez. Para um problema com n -cidades, a solução final, descrita pelo vetor \mathbf{p} , deve representar um possível percurso em que o valor do elemento p_i é a posição da i -ésima cidade no respectivo percurso. Por exemplo, se para um problema do caixeiro viajante com 4 cidades a solução final é representada pelo caminho definido por: cidade 4, cidade 1, cidade 3, cidade 2; então o vetor \mathbf{p} representando este percurso será dado por:

$$\mathbf{p}^T = [2 \ 4 \ 3 \ 1]$$

onde a posição da cidade 'i', na lista de visitas do percurso, é dado pelo respectivo valor do elemento p_i . Para o exemplo citado, a posição da cidade 1 na lista de visita vale 2, ou seja, a cidade 1 seria a segunda cidade a ser visitada.

Pode-se representar o vetor \mathbf{p} por um vetor \mathbf{v} (saída da rede), com elementos definidos por zero ou um, utilizando produtos de Kronecker (produto tensor). Na notação utilizando produtos de Kronecker, têm-se as seguintes definições [Grah81]:

- δ é uma matriz de dimensão $n \times n$ ($\delta \in \mathfrak{R}^{n \times n}$) definida por:

$$\delta_{ij} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} \tag{5.2}$$

e $\delta(k) \in \mathfrak{R}^n$ é o vetor coluna correspondente à k -ésima coluna de δ .

- $\mathbf{v}(\mathbf{p})$ é um vetor de dimensão $n.m$ ($\mathbf{v}(\mathbf{p}) \in \mathfrak{R}^{n.m}$) que representa a forma final do vetor de saída da rede, \mathbf{v} , correspondente à solução do problema representado por \mathbf{p} . O vetor $\mathbf{v}(\mathbf{p})$ é definido por:

$$\mathbf{v}(\mathbf{p}) = \begin{bmatrix} \delta(p_1) \\ \delta(p_2) \\ \vdots \\ \delta(p_n) \end{bmatrix} \tag{5.3}$$

- $\mathbf{V}(\mathbf{p})$ é uma matriz de dimensão $n \times m$ ($\mathbf{V}(\mathbf{p}) \in \mathfrak{R}^{n \times m}$) que corresponde a uma forma alternativa de se representar o vetor $\mathbf{v}(\mathbf{p})$. Esta matriz é definida por:

As propriedades dos produtos de Kronecker que serão utilizadas neste capítulo são as seguintes [Grah81]:

$$(\lambda \mathbf{w} \otimes \gamma \mathbf{h}) = \lambda \gamma (\mathbf{w} \otimes \mathbf{h}) \quad (5.8)$$

$$(\mathbf{w} \otimes \mathbf{h})^T (\mathbf{x} \otimes \mathbf{g}) = (\mathbf{w}^T \mathbf{x}) (\mathbf{h}^T \mathbf{g}) \quad (5.9)$$

$$(\mathbf{P} \otimes \mathbf{Q})(\mathbf{w} \otimes \mathbf{h}) = (\mathbf{P}\mathbf{w} \otimes \mathbf{Q}\mathbf{h}) \quad (5.10)$$

$$(\mathbf{P}^T \otimes \mathbf{Q}^T) = (\mathbf{P} \otimes \mathbf{Q})^T \quad (5.11)$$

$$(\mathbf{P} \otimes \mathbf{Q})(\mathbf{E} \otimes \mathbf{F}) = (\mathbf{P}\mathbf{E} \otimes \mathbf{Q}\mathbf{F}) \quad (5.12)$$

$$\text{traço}(\mathbf{V}^T \cdot \mathbf{E}) = \text{vec}(\mathbf{V})^T \cdot \text{vec}(\mathbf{E}) \quad (5.13)$$

$$\text{vec}(\mathbf{Q} \cdot \mathbf{V} \cdot \mathbf{P}^T) = (\mathbf{P} \otimes \mathbf{Q}) \cdot \text{vec}(\mathbf{V}) \quad (5.14)$$

$$\text{traço}(\mathbf{V}^T \cdot \mathbf{Q} \cdot \mathbf{V} \cdot \mathbf{P}^T) = \text{vec}(\mathbf{V})^T \cdot (\mathbf{P} \otimes \mathbf{Q}) \cdot \text{vec}(\mathbf{V}) \quad (5.15)$$

- \mathbf{o}^n e \mathbf{O}^n são respectivamente um vetor de dimensão 'n' e uma matriz de dimensão $n \times n$ com todos os seus elementos possuindo valor unitário, ou seja:

$$\left. \begin{array}{l} [\mathbf{o}]_i = 1 \\ [\mathbf{O}]_{ij} = 1 \end{array} \right\} \text{para } i, j \in \{1, \dots, n\} \quad (5.16)$$

- \mathbf{R}^n é uma matriz projeção (i.e. $\mathbf{R}^n \cdot \mathbf{R}^n = \mathbf{R}^n$) de dimensão $n \times n$, tal que a pré-multiplicação de uma matriz por \mathbf{R}^n transforma a soma dos elementos de cada coluna dessa matriz para zero, enquanto a pós-multiplicação transforma a soma dos elementos de cada linha para zero. A matriz \mathbf{R}^n é definida por:

$$\mathbf{R}^n = \mathbf{I}^n - \frac{1}{n} \mathbf{O}^n \quad (5.17)$$

As propriedades utilizadas no mapeamento dos problemas de programação combinatorial e programação dinâmica, são dadas por:

- Vetor soma de cada uma das colunas de uma matriz \mathbf{V} , denotado por \mathbf{c} , é dado por:

$$\begin{aligned} c_i &= \sum_{x=1}^n V_{xi} \\ \mathbf{c}^T &= \mathbf{o}^{n^T} \cdot \mathbf{V} \end{aligned} \quad (5.18)$$

- Vetor soma de cada uma das linhas de \mathbf{V} , denotado por \mathbf{l} , é dado por:

$$\begin{aligned} l_i &= \sum_{x=1}^m V_{xi} \\ \mathbf{l}^T &= \mathbf{V} \cdot \mathbf{o}^m \end{aligned} \quad (5.19)$$

• Pré-multiplicação de uma dada matriz $\mathbf{A}^{m \times n}$ por \mathbf{R}^m transforma a soma dos elementos de cada coluna dessa matriz para zero, ou seja:

$$\begin{aligned} \sum_{i=1}^m [\mathbf{R}^m \cdot \mathbf{a}]_i &= \sum_{i=1}^m \left[\mathbf{a} - \frac{1}{m} \mathbf{O}^m \cdot \mathbf{a} \right]_i \\ &= \mathbf{o}^{m^T} \cdot \left[\mathbf{a} - \frac{1}{m} \mathbf{O}^m \cdot \mathbf{a} \right] \\ &= \mathbf{o}^{m^T} \cdot \mathbf{a} - \frac{1}{m} (\mathbf{o}^{m^T} \cdot \mathbf{O}^m \cdot \mathbf{o}^{m^T}) \cdot \mathbf{a} = \mathbf{o}^{m^T} \cdot \mathbf{a} - \mathbf{o}^{m^T} \cdot \mathbf{a} \\ &= \mathbf{o}^{m^T} \cdot \mathbf{a} - \mathbf{o}^{m^T} \cdot \mathbf{a} = 0 \end{aligned}$$

onde o vetor \mathbf{a} corresponde a um dos vetores coluna da matriz \mathbf{A} .

• Pós-multiplicação de uma dada matriz $\mathbf{A}^{m \times n}$ por \mathbf{R}^n transforma a soma dos elementos de cada linha dessa matriz para zero, ou seja:

$$\begin{aligned} \sum_{i=1}^n [\mathbf{a}^T \cdot \mathbf{R}^n]_i &= \sum_{i=1}^n \left[\mathbf{a}^T - \frac{1}{n} \mathbf{a}^T \cdot \mathbf{O}^n \right]_i \\ &= \left[\mathbf{a}^T - \frac{1}{n} \mathbf{a}^T \cdot \mathbf{O}^n \right] \cdot \mathbf{o}^n \\ &= \mathbf{a}^T \cdot \mathbf{o}^n - \frac{1}{n} \mathbf{a}^T \cdot (\mathbf{o}^n \cdot \mathbf{O}^{n^T} \cdot \mathbf{o}^n) \cdot \mathbf{a} = \mathbf{a}^T \cdot \mathbf{o}^n - \mathbf{a}^T \cdot \mathbf{o}^n \\ &= \mathbf{a}^T \cdot \mathbf{o}^n - \mathbf{a}^T \cdot \mathbf{o}^n = 0 \end{aligned}$$

onde o vetor \mathbf{a} corresponde a um dos vetores linha da matriz \mathbf{A} .

Para ilustrar algumas destas propriedades, um exemplo em três dimensões é dado a seguir.

Exemplo em Três Dimensões

Um subespaço-válido para um problema específico, onde existe um conjunto finito de soluções, pode ser formulado com a finalidade de agrupar somente as soluções consideradas válidas. Uma ilustração simplificada deste princípio básico pode ser observado na Figura 5.1, onde se considera duas formas possíveis de solução válida: o conjunto definido por

$$\{(1,0,0), (0,1,0), (0,0,1)\} \tag{5.20}$$

ou o conjunto

$$\{(1,1,0), (1,0,1), (0,1,1)\} \tag{5.21}$$

Na Figura 5.1, o plano hachurado mais claro (plano (a)) é um subespaço-válido que contém apenas as soluções do primeiro conjunto, enquanto o plano hachurado mais escuro (plano (b)) contém somente as soluções do segundo conjunto. Definindo S como sendo a soma dos elementos de cada vetor pertencente a cada um dos conjuntos, ou seja:

$$S = \sum_{k=1}^3 v_k \tag{5.22}$$

uma forma de visualizar o plano (a) é aquela onde $S = 1$, enquanto no plano 2, o valor de S é $S = 2$. Assim, se $S = 1$, o vetor v pertence ao primeiro conjunto de soluções (5.20); e se $S = 2$, o vetor v deve pertencer ao segundo conjunto (5.21).

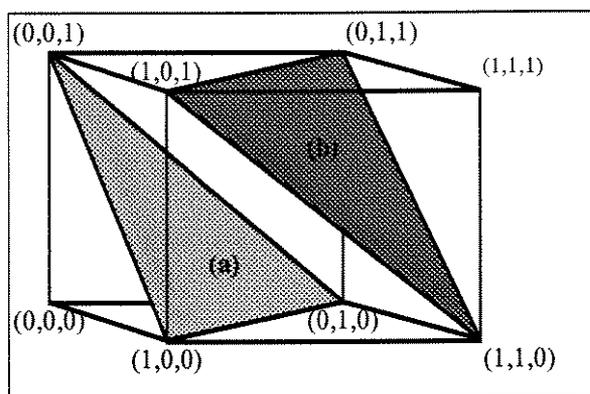


Figura 5.1. Ilustração do Subespaço-válido

Utilizando as definições de \mathbf{o}^n e \mathbf{R}^n , apresenta-se a seguir a forma analítica do subespaço-válido para o exemplo. Uma solução válida para o primeiro conjunto de soluções é aquela onde $S = 1$, e uma forma equivalente para representar uma solução válida deste conjunto pode ser definida por:

$$\mathbf{x} = \frac{1}{3} \mathbf{o}^3 \tag{5.23}$$

Neste caso, todas as soluções do primeiro conjunto, dadas por (5.20), podem ser representadas pela equação (5.23), que corresponderá ao vetor independente \mathbf{s} na equação do subespaço-válido. Deste modo, a dedução de \mathbf{T}^{val} é realizada utilizando os seguintes procedimentos:

$$\begin{aligned} \mathbf{v} &= \mathbf{T}^{val} \cdot \mathbf{v} + \mathbf{s} \\ \mathbf{v} &= \mathbf{T}^{val} \cdot \mathbf{v} + \frac{1}{3} \mathbf{o}^3 \\ \mathbf{T}^{val} \cdot \mathbf{v} &= \mathbf{v} - \frac{1}{3} \mathbf{o}^3 \end{aligned} \tag{5.24}$$

Pós-multiplicando ambos os membros da equação (5.24) por $\mathbf{v}^T \cdot \mathbf{o}^3$, tem-se:

$$\begin{aligned}
 \mathbf{T}^{\text{val}} \cdot \mathbf{v} \cdot \mathbf{v}^T \cdot \mathbf{o}^3 &= \mathbf{v} \cdot \mathbf{v}^T \cdot \mathbf{o}^3 - \frac{1}{3} \mathbf{o}^3 \cdot \mathbf{v}^T \cdot \mathbf{o}^3 \\
 \mathbf{T}^{\text{val}} \cdot \mathbf{v} \cdot \mathbf{v}^T \cdot \mathbf{o}^3 &= \mathbf{v} \cdot \mathbf{v}^T \cdot \mathbf{o}^3 - \frac{1}{3} \mathbf{o}^3 \cdot \mathbf{o}^{3T} \cdot \mathbf{v} \\
 \mathbf{T}^{\text{val}} \cdot \mathbf{v} \cdot \mathbf{v}^T \cdot \mathbf{o}^3 &= \mathbf{v} \cdot \mathbf{v}^T \cdot \mathbf{o}^3 - \frac{1}{3} \mathbf{O}^3 \cdot \mathbf{v}
 \end{aligned} \tag{5.25}$$

como $\mathbf{o}^{3T} \cdot \mathbf{v} = \mathbf{v}^T \cdot \mathbf{o}^3 = S = 1$, a expressão explícita para \mathbf{T}^{val} obtida de (5.25) torna-se:

$$\begin{aligned}
 \mathbf{T}^{\text{val}} \cdot \mathbf{v} &= \mathbf{v} - \frac{1}{3} \mathbf{O}^3 \cdot \mathbf{v} \\
 \mathbf{T}^{\text{val}} \cdot \mathbf{v} &= [\mathbf{I}^3 - \frac{1}{3} \mathbf{O}^3] \cdot \mathbf{v} \\
 \mathbf{T}^{\text{val}} &= [\mathbf{I}^3 - \frac{1}{3} \mathbf{O}^3] \\
 \mathbf{T}^{\text{val}} &= \mathbf{R}^3
 \end{aligned} \tag{5.26}$$

A equação analítica do subespaço-válido, que descreve todas as soluções pertencentes ao primeiro conjunto de soluções válidas, é dada por:

$$\mathbf{v} = \mathbf{R}^3 \cdot \mathbf{v} + \frac{1}{3} \mathbf{o}^3 \tag{5.27}$$

onde o termo $\mathbf{R}^3 \cdot \mathbf{v}$ da equação (5.27) realiza a projeção do vetor \mathbf{v} no subespaço ortogonal a \mathbf{o}^3 . Para obtermos $S = 2$, representação do segundo conjunto de soluções válidas dada por (5.21), o termo $\frac{2}{3} \mathbf{o}^3$ assegura que este valor sempre será igual a 2, e então tem-se que:

$$\mathbf{v} = \mathbf{R}^3 \cdot \mathbf{v} + \frac{2}{3} \mathbf{o}^3 \tag{5.28}$$

é a equação do subespaço-válido que descreve todas as soluções pertencentes ao segundo conjunto de soluções.

5.3-) PROBLEMAS DE OTIMIZAÇÃO COMBINATORIAL

Problemas de otimização combinatorial ocorrem naturalmente em áreas de ciências e engenharia, quando é necessário otimizar uma função custo (objetivo) num espaço discreto ou enumerável. A procura da solução exata de uma grande classe destes problemas requer um esforço computacional que cresce exponencialmente com o tamanho do problema. Métodos computacionalmente tratáveis têm sido desenvolvidos através de heurísticas, que são essencialmente um conjunto de procedimentos que geralmente permitem encontrar soluções factíveis em um tempo de busca razoável. Como um exemplo ilustrativo, para o problema do

caixeiro viajante com n -cidades, o número de caminhos possíveis é dado por $\frac{n!}{2n}$, e para $n = 30$ tem-se 4.42×10^{30} caminhos possíveis.

Esta dificuldade em encontrar a solução ótima entre um conjunto de soluções possíveis é uma característica de uma classe de problemas de otimização combinatorial que são conhecidos como NP-completo [Gare79]. Para estes problemas nenhum algoritmo conhecido garante encontrar a solução ótima em tempo polinomial [Papa82]. Pesquisas recentes desenvolveram novos algoritmos baseados em diversas heurísticas derivadas de diferentes fontes do conhecimento. Entre as técnicas mais utilizadas destacam-se a 'Busca Tabu' [Glov90], que se caracteriza por restringir certas partes do espaço de busca durante algum tempo, direcionando a busca para determinadas áreas; o 'Simulated Annealing' [Kirk83], que é baseado na mecânica estatística envolvida com o resfriamento da matéria; e os 'Algoritmos Genéticos' [Davi90], que são baseados na utilização de conceitos derivados da evolução Darwiniana que modifica e seleciona populações de soluções candidatas. Estas técnicas tem sido aplicadas satisfatoriamente a uma grande variedade de problemas NP-completo.

Nesta seção, mostra-se como a rede de Hopfield modificada, desenvolvida no Capítulo 2, pode ser utilizada para resolver problemas de otimização combinatorial. Em particular, mapeia-se o problema do caixeiro viajante, das N -rainhas e do matching-bipartido, através da RHM.

5.3.1-) Mapeamento do Problema do Caixeiro Viajante

A seguir são deduzidos os parâmetros (T^{val} , s , T^{ot} e i^{ot}) da rede de Hopfield modificada para o problema do caixeiro viajante. Como descrito anteriormente, o problema consiste em encontrar o menor caminho entre n -cidades que devem ser visitadas uma única vez.

Dedução das Equações Para T^{val} e s

Inicialmente, se v for uma solução válida $v(p)$, então a soma das linhas e colunas de V serão dadas por c^T e l^T , definidas respectivamente em (5.18) e (5.19). Para o problema do caixeiro viajante, $V(p)$ é uma matriz quadrada ($n = m$), onde apenas um elemento de cada linha ou coluna possui valor igual a 1, ou seja:

$$[\mathbf{V}(\mathbf{p})]_{ij} \in \{1,0\}$$

$$\sum_{i=1}^n [\mathbf{V}(\mathbf{p})]_{ij} = \sum_{j=1}^n [\mathbf{V}(\mathbf{p})]_{ij} = 1 \quad (i, j \in \{1..n\}) \quad (5.29)$$

Para esta classe de problemas, dado que $\mathbf{v} = \text{vec}(\mathbf{V}^T)$ é uma solução válida, a condição $\mathbf{c} = \mathbf{1} = \mathbf{o}^n$ é suficiente para garantir que \mathbf{v} é da forma $\mathbf{v}(\mathbf{p})$. Assim um subespaço $\mathbf{S} \in \mathcal{R}^{n \times n}$ para este problema pode ser representado da seguinte forma:

$$\mathbf{S} = \mathbf{V} = \frac{1}{n} \mathbf{o}^n \cdot \mathbf{o}^{nT} \quad (5.30)$$

Neste caso, a matriz \mathbf{T}^{val} do subespaço-válido deve garantir que a soma de cada linha ou coluna da matriz \mathbf{V} seja igual a zero. Esta operação é efetuada utilizando-se as propriedades da matriz \mathbf{R}^n dada em (5.17), isto é, pré-multiplicando e pós-multiplicando \mathbf{V} por \mathbf{R}^n , tem-se:

$$\mathbf{R}^n \cdot \mathbf{V} \cdot \mathbf{R}^n = \mathbf{T}^{\text{val}} \cdot \mathbf{V} \quad (5.31)$$

Substituindo-se (5.30) e (5.31) na equação do subespaço-válido, dada por (2.8), tem-se:

$$\mathbf{V} = \mathbf{R}^n \cdot \mathbf{V} \cdot \mathbf{R}^n + \frac{1}{n} \mathbf{o}^n \cdot \mathbf{o}^{nT} \quad (5.32)$$

Aplicando-se o operador $\text{vec}(\cdot)$ dado em (5.14) na equação (5.32), tem-se:

$$\begin{aligned} \text{vec}(\mathbf{V}) &= \text{vec}(\mathbf{R}^n \cdot \mathbf{V} \cdot \mathbf{R}^n) + \frac{1}{n} \text{vec}(\mathbf{o}^n \cdot \mathbf{o}^{nT}) \\ \text{vec}(\mathbf{V}) &= (\mathbf{R}^n \otimes \mathbf{R}^n) \cdot \text{vec}(\mathbf{V}) + \frac{1}{n} (\mathbf{o}^n \otimes \mathbf{o}^n) \end{aligned} \quad (5.33)$$

Substituindo-se $\text{vec}(\mathbf{V})$ por \mathbf{v} na equação (5.33), tem-se:

$$\mathbf{v} = (\mathbf{R}^n \otimes \mathbf{R}^n) \cdot \mathbf{v} + \frac{1}{n} (\mathbf{o}^n \otimes \mathbf{o}^n) \quad (5.34)$$

Por identidade com a equação do subespaço-válido ($\mathbf{v} = \mathbf{T}^{\text{val}} \cdot \mathbf{v} + \mathbf{s}$), conclui-se que os parâmetros \mathbf{T}^{val} e \mathbf{s} para o problema do caixeiro viajante são dados por:

$$\mathbf{T}^{\text{val}} = (\mathbf{R}^n \otimes \mathbf{R}^n) \quad (5.35)$$

$$\mathbf{s} = \frac{1}{n} (\mathbf{o}^n \otimes \mathbf{o}^n) \quad (5.36)$$

As equações (5.35) e (5.36) obedecem as propriedades do subespaço-válido, isto é, $\mathbf{T}^{\text{val}} \cdot \mathbf{T}^{\text{val}} = \mathbf{T}^{\text{val}}$ e $\mathbf{T}^{\text{val}} \cdot \mathbf{s} = \mathbf{0}$.

Dedução das Equações Para T^{ot} e i^{ot}

A fim de se obter uma equação, envolvendo $V(\mathbf{p})$, para o comprimento do percurso especificado pela ordem dos elementos do vetor \mathbf{p} (ordem das cidades), define-se a matriz \mathbf{P} de dimensão $n \times n$ como sendo a matriz cujos elementos representam a distância entre as cidades a serem visitadas, ou seja:

$$[\mathbf{P}]_{ij} \equiv \text{distância entre a cidade 'i' e 'j'}$$

Definindo também a matriz \mathbf{Q} de dimensão $n \times n$ como sendo a matriz cujos elementos representam o encadeamento entre as cidades a serem visitadas [Hopf85], ou seja:

$$[\mathbf{Q}]_{ij} = \delta_{j-1,i} + \delta_{j+1,i} \quad (i, j \in \{1..n\})$$

O comprimento total do percurso correspondente ao vetor ordenado \mathbf{p} é dado por:

$$d(\mathbf{p}) = \frac{1}{2} \text{traço}[\mathbf{V}(\mathbf{p}).\mathbf{P}.\mathbf{V}(\mathbf{p})^T.\mathbf{Q}] \tag{5.37}$$

Os parâmetros T^{ot} e i^{ot} , correspondentes ao termo de energia E^{ot} , podem ser obtidos fazendo-se $E^{ot} = d(\mathbf{p})$. Portanto, a rede executará a otimização desejada (minimizar $d(\mathbf{p})$), ou seja:

$$\begin{aligned} E^{ot} &= d(\mathbf{p}) \\ &= \frac{1}{2} \text{traço}[\mathbf{V}(\mathbf{p}).\mathbf{P}.\mathbf{V}(\mathbf{p})^T.\mathbf{Q}] \\ &= \frac{1}{2} \text{traço}[\mathbf{V}(\mathbf{p})^T.\mathbf{Q}.\mathbf{V}(\mathbf{p}).\mathbf{P}] \\ &= \frac{1}{2} [\text{vec}(\mathbf{V}(\mathbf{p}))]^T.\text{vec}(\mathbf{Q}.\mathbf{V}(\mathbf{p}).\mathbf{P}) \\ &= \frac{1}{2} [\text{vec}(\mathbf{V}(\mathbf{p}))]^T.(\mathbf{P} \otimes \mathbf{Q}).\text{vec}(\mathbf{V}(\mathbf{p})) \\ &= \frac{1}{2} \mathbf{v}(\mathbf{p})^T.(\mathbf{P} \otimes \mathbf{Q}).\mathbf{v}(\mathbf{p}) \end{aligned} \tag{5.38}$$

Comparando-se a equação (5.38) com a equação do termo E^{ot} dada em (2.6), obtêm-se os parâmetros T^{ot} e i^{ot} definidos por:

$$T^{ot} = -(\mathbf{P} \otimes \mathbf{Q}) \tag{5.39}$$

$$i^{ot} = \mathbf{0} \tag{5.40}$$

Os valores dos elementos do vetor de saída da rede $\mathbf{v}(\mathbf{p})$ são limitados entre zero e um, através da função de ativação rampa-simétrica.

Resultados de Simulação

O mapeamento proposto acima é utilizado na solução do problema do caixeiro viajante para 30 cidades. Os resultados obtidos pela rede de Hopfield modificada (RHM) são comparados com os resultados obtidos através de 4 abordagens que geralmente são utilizadas na resolução deste tipo de problema [Durb87]: a rede de Hopfield convencional [Hopf85] (descrita brevemente no Apêndice C), a Elastic Net [Durb87], a técnica do Simulated Annealing [Kirk83], e a heurística 'Nearest Neighbour' [Papa82]. O desempenho de cada abordagem é representado pelo comprimento mínimo de percurso obtido na resolução do problema do caixeiro viajante para 30 cidades. As coordenadas das cidades são idênticas àquelas utilizadas por Hopfield e Tank em [Hopf85], cuja solução ótima (menor percurso) vale 4.26.

Abordagem	Percurso
Hopfield e Tank	5.07
Elastic Net	4.27
Simulated Annealing	4.31
Nearest Neighbour	4.75
RHM	4.30

Tabela 5.1. Comparação dos Resultados (Prob. do Caixeiro Viajante)

Os resultados obtidos com os vários métodos são apresentados na Tabela 5.1. Nota-se que a solução obtida pela rede de Hopfield modificada possui um erro menor que 1% comparado à solução ótima com valor de 4.26 [Hopf85]. A qualidade dos resultados é semelhante aos obtidos pela Elastic Net proposto por [Durb87] e também aos da técnica do Simulated Annealing [Kirk83], sendo sensivelmente melhor que a heurística Nearest Neighbour (vizinho mais próximo) e que os resultados obtidos pela rede de Hopfield convencional.

A rede de Hopfield modificada converge em cerca de 1.500 iterações, enquanto que a rede de Hopfield convencional converge em cerca de 20.000 iterações. Esta aceleração de convergência ocorre devido ao controlador nebuloso que calcula o valor de Δt em cada iteração. Outro fato que contribui para a aceleração da convergência da RHM é que a minimização do termo de energia E^{conf} não interfere na minimização do termo E^{ot} , conseqüentemente, as simulações por computador da rede de Hopfield modificada convergem com um menor número de iterações em relação a rede de Hopfield convencional. Além disso, a rede de Hopfield modificada tem a vantagem adicional de não requerer o ajuste de nenhum

parâmetro de inicialização, enquanto que, na rede de Hopfield convencional, é necessário o ajuste de 5 parâmetros (A, B, C, D e Δt) como descrito em [Hopf85] e mostrado no Apêndice C.7. Isto pode aumentar excessivamente o tempo de computação da rede, visto que A, B, C e D são os parâmetros que ponderam as restrições impostas pelo problema.

5.3.2-) Mapeamento do Problema das N-Rainhas

O problema das N-rainhas é um típico problema que pode ser resolvido por otimização combinatorial. Este problema foi estudado por C. F. Gauss em 1850 e é enunciado a seguir [Wirt76]: N-rainhas devem ser colocadas em um tabuleiro de xadrez de dimensão NxN, de modo que nenhuma delas ataque ou seja atacada por nenhuma outra. Um modo, pouco viável, de se resolver este problema é através da busca exaustiva, isto é, enumeram-se todas as combinações possíveis e utiliza-se um método de tentativa e erro. O número de combinações possíveis (NCP^{Queen}) é dado por:

$$NCP^{Queen} = \frac{N^2!}{N!(N^2 - N)!} \quad (5.41)$$

e para $N = 8$, tem-se $4,426 \times 10^9$ combinações possíveis. A pesquisa de todas estas combinações torna o algoritmo de busca exaustiva pouco atrativo, pois exige-se um enorme esforço computacional.

A utilização de redes neurais para resolver o problema das N-rainhas também foi proposto em [Mand92]. Uma breve descrição desta rede é apresentada no Apêndice C. Todavia, a rede desenvolvida neste artigo frequentemente encontra um ponto de equilíbrio que não satisfaz nenhuma solução factível do problema. Uma das principais razões para este fato é a dificuldade em especificar os valores corretos para as diversas constantes de ponderações relativos aos termos de restrições [Hedg88]. Assim, de acordo com as constantes escolhidas, a interferência entre os termos de restrições e o termo de otimização gera soluções infactíveis. Com o propósito de contornar este problema, aplica-se a rede de Hopfield modificada, desenvolvida na Seção 2.4, para resolver este problema de otimização combinatorial.

A partir das regras do xadrez, sabe-se que a rainha pode atacar peças que se encontrem na mesma linha, coluna ou diagonal do tabuleiro, logo pode-se deduzir inicialmente, que cada linha e cada coluna deve conter apenas uma rainha. Portanto, dado que cada casa (quadrado) do tabuleiro pode ser representada adequadamente por um neurônio; as restrições estruturais

para o problema, minimizadas pelo termo de energia E^{conf} , e que são mapeadas pelo subespaço-válido são dadas por:

- Cada linha deve conter somente uma rainha.
- Cada coluna deve conter somente uma rainha.

A minimização do termo de energia E^{ot} corresponderá, portanto, a minimização do número de rainhas em cada uma das diagonais do tabuleiro de xadrez. Deve-se notar que existem várias soluções válidas (igualmente ótimas) para o problema específico das N-rainhas, e a determinação de uma destas soluções depende exclusivamente dos valores atribuídos, para v , na inicialização da rede. Esta condição difere do problema do caixeiro viajante em que apenas uma solução ótima deve ser obtida pela rede.

A implementação do problema das N-rainhas, como definido anteriormente, pode ser feito a partir das definições apresentadas na Seção 5.2. Define-se o vetor \mathbf{p} como sendo o vetor que contém os índices das casas onde serão colocadas as rainhas nas diversas linhas do tabuleiro. Portanto, o valor do elemento p_i representa que casa estará ativada (ocupada por uma rainha) na i -ésima linha do tabuleiro, ou seja:

$$p_i \in \{1, \dots, N\} \text{ onde } i \in \{1 .. N\}$$

sendo ‘N’ o número de rainhas para o problema. O vetor $\mathbf{v}(\mathbf{p})$ e $\mathbf{V}(\mathbf{p})$ são dados por:

$$\mathbf{v}(\mathbf{p}) = \begin{bmatrix} \delta(p_1) \\ \delta(p_2) \\ \delta(p_3) \\ \vdots \\ \delta(p_n) \end{bmatrix}_{N \times N} \qquad \mathbf{V}(\mathbf{p}) = \begin{bmatrix} \delta(p_1)^T \\ \delta(p_2)^T \\ \delta(p_3)^T \\ \vdots \\ \delta(p_n)^T \end{bmatrix}_{N \times N}$$

onde $\delta(k) \in \mathcal{R}^N$ é um vetor coluna correspondente a k -ésima coluna de δ .

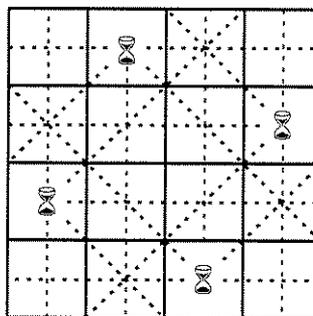


Figura 5.2. O Problema das N-Rainhas

Como ilustração, para um problema das N-rainhas com $N = 4$, cuja representação gráfica de uma das soluções é mostrada na Figura 5.2, os vetores \mathbf{p} e $\mathbf{v}(\mathbf{p})$, e a matriz $\mathbf{V}(\mathbf{p})$ são dados por:

$$\mathbf{p}^T = [2 \quad 4 \quad 1 \quad 3]$$

$$\mathbf{v}(\mathbf{p})^T = [0 \ 1 \ 0 \ 0 \quad 0 \ 0 \ 0 \ 1 \quad 1 \ 0 \ 0 \ 0 \quad 0 \ 0 \ 1 \ 0]$$

$$\delta(\mathbf{p}_1)^T \quad \delta(\mathbf{p}_2)^T \quad \delta(\mathbf{p}_3)^T \quad \delta(\mathbf{p}_4)^T$$

$$\mathbf{V}(\mathbf{p}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Deduz-se a seguir, as equações para os parâmetros internos (\mathbf{T}^{val} , \mathbf{s} , \mathbf{T}^{ot} e \mathbf{i}^{ot}) da rede de Hopfield modificada para o problema das N-rainhas.

Dedução das Equações Para \mathbf{T}^{val} e \mathbf{s}

As equações de \mathbf{T}^{val} e \mathbf{s} do subespaço-válido para o problema das N-rainhas são desenvolvidas utilizando os conceitos descritos na Seção 5.2, e são semelhantes àquelas do problema do caixeiro viajante. As restrições estruturais para o problema das N-rainhas mostram que apenas um neurônio por linha ou coluna deve estar ativado. Esta restrição significa que apenas um e somente um elemento de cada linha ou coluna da matriz $\mathbf{V}(\mathbf{p})$ deve possuir valor igual a 1, ou seja:

$$[\mathbf{V}(\mathbf{p})]_{ij} \in \{1,0\}$$

$$\sum_{i=1}^N [\mathbf{V}(\mathbf{p})]_{ij} = \sum_{j=1}^N [\mathbf{V}(\mathbf{p})]_{ij} = 1 \quad (i, j \in \{1..N\}) \tag{5.42}$$

Para este problema, dado que $\mathbf{v} = \text{vec}(\mathbf{V}^T)$ é uma solução válida, um subespaço para este problema pode ser representado da seguinte forma:

$$\mathbf{S} = \mathbf{V} = \frac{1}{N} \mathbf{o}^N \cdot \mathbf{o}^{N^T} \tag{5.43}$$

Neste caso, para que a solução seja válida durante toda a evolução da rede, a matriz \mathbf{T}^{val} do subespaço-válido deve garantir que a soma de cada linha ou coluna da matriz \mathbf{V} seja igual a

zero. Esta operação é efetuada, utilizando-se as propriedades da matriz \mathbf{R}^n dada em (5.17), isto é:

$$\mathbf{R}^N \cdot \mathbf{V} \cdot \mathbf{R}^N = \mathbf{T}^{val} \cdot \mathbf{V} \tag{5.44}$$

Substituindo-se (5.43) e (5.44) na equação do subespaço-válido, tem-se:

$$\mathbf{V} = \mathbf{R}^N \cdot \mathbf{V} \cdot \mathbf{R}^N + \frac{1}{N} \mathbf{o}^N \cdot \mathbf{o}^{N^T} \tag{5.45}$$

Aplicando-se o operador $\text{vec}(\cdot)$ dado em (5.14) na equação (5.45), obtém-se:

$$\begin{aligned} \text{vec}(\mathbf{V}) &= \text{vec}(\mathbf{R}^N \cdot \mathbf{V} \cdot \mathbf{R}^N) + \frac{1}{N} \text{vec}(\mathbf{o}^N \cdot \mathbf{o}^{N^T}) \\ \text{vec}(\mathbf{V}) &= (\mathbf{R}^N \otimes \mathbf{R}^N) \cdot \text{vec}(\mathbf{V}) + \frac{1}{N} (\mathbf{o}^N \otimes \mathbf{o}^N) \end{aligned} \tag{5.46}$$

Substituindo-se $\text{vec}(\mathbf{V})$ por \mathbf{v} na equação (5.46), tem-se:

$$\mathbf{v} = (\mathbf{R}^N \otimes \mathbf{R}^N) \cdot \mathbf{v} + \frac{1}{N} (\mathbf{o}^N \otimes \mathbf{o}^N)$$

Por identidade com a equação do subespaço-válido ($\mathbf{v} = \mathbf{T}^{val} \cdot \mathbf{v} + \mathbf{s}$), resulta que os parâmetros \mathbf{T}^{val} e \mathbf{s} para o problema das N-rainhas são dados por:

$$\mathbf{T}^{val} = (\mathbf{R}^N \otimes \mathbf{R}^N) \tag{5.47}$$

$$\mathbf{s} = \frac{1}{N} (\mathbf{o}^N \otimes \mathbf{o}^N) \tag{5.48}$$

As equações (5.47) e (5.48) satisfazem as propriedades do subespaço-válido, isto é, $\mathbf{T}^{val} \cdot \mathbf{T}^{val} = \mathbf{T}^{val}$ e $\mathbf{T}^{val} \cdot \mathbf{s} = \mathbf{0}$.

Dedução das Equações Para \mathbf{T}^{ot} e \mathbf{i}^{ot}

O termo de energia E^{ot} da rede de Hopfield modificada para o problema das N-rainhas é escolhido com o objetivo de que a solução ótima corresponda ao menor número de rainhas em cada uma das diagonais do tabuleiro de xadrez. Quando o termo de energia E^{ot} for minimizado, a solução ótima corresponde a um dos estados de menor energia da rede. O termo E^{ot} para o problema das N-rainhas é definido por:

$$E^{ot} = \underbrace{\frac{1}{2} \sum_{i=2}^N \sum_{j=1}^{i-1} [(\sum_{\substack{k=i-j+1 \\ k \neq i}}^N v_{k,k-i+j}) \cdot v_{ij}]}_{1o. \text{ Termo}} + \underbrace{\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N [(\sum_{\substack{k=1 \\ k \neq i}}^{N+i-j} v_{k,k-i+j}) \cdot v_{ij}]}_{2o. \text{ Termo}} + \underbrace{\frac{1}{2} \sum_{i=1}^N \sum_{j=N-i+1}^N [(\sum_{\substack{k=i+j-N \\ k \neq i}}^N v_{k,i+j-k}) \cdot v_{ij}]}_{3o. \text{ Termo}} + \underbrace{\frac{1}{2} \sum_{i=1}^{N-1} \sum_{j=1}^{i+j-1} [(\sum_{\substack{k=1 \\ k \neq i}}^N v_{k,i+j-k}) \cdot v_{ij}]}_{4o. \text{ Termo}} \tag{5.49}$$

Na equação (5.49), o primeiro termo fornece o produto entre cada elemento v_{ij} e todos os outros elementos que estão na diagonal direita inferior, enquanto o segundo termo fornece o produto de cada elemento v_{ij} com todos os outros elementos que estão na diagonal esquerda superior. Por conseguinte, o terceiro termo fornece o produto entre cada elemento v_{ij} e todos os outros elementos que estão na diagonal direita superior, e o quarto termo fornece o produto de cada elemento v_{ij} com todos os outros elementos que estão na diagonal esquerda inferior.

Derivando-se a equação (5.49) na direção oposta ao gradiente de E^{ot} , em relação a v_{ij} , tem-se:

$$-\frac{\partial E^{ot}}{\partial v_{ij}} = -\left[\sum_{\substack{k=i-j+1 \\ k \neq i > j}}^N v_{k,k-i+j} + \sum_{\substack{k=1 \\ k \neq i \leq j}}^{N+i-j} v_{k,k-i+j} + \sum_{\substack{k=i+j-N \\ k \neq i > N-j}}^N v_{k,i+j-k} + \sum_{\substack{k=i \\ k \neq i \leq N-j}}^{i+j-1} v_{k,i+j-k} \right] \quad (5.50)$$

Desenvolvendo a equação (5.50), obtêm-se os parâmetros T^{ot} e i^{ot} da rede para o problema das N-rainhas. Utilizando-se produto de Kronecher, estes parâmetros são definidos por:

$$T^{ot} = -[(Q_1^{ot} \otimes Q_1^{ot}) + (Q_2^{ot} \otimes Q_2^{ot}) + \dots + (Q_{(N-1)}^{ot} \otimes Q_{(N-1)}^{ot})] \quad (5.51)$$

$$i^{ot} = 0 \quad (5.52)$$

onde a matriz $Q_k^{ot} \in \mathcal{R}^{N \times N}$ é dada por:

$$[Q_k^{ot}]_{ij} = \delta_{j-k,i} + \delta_{j+k,i} \quad (5.53)$$

Como ilustração, para $N = 4$, a componente Q_2^{ot} do segundo termo da equação de T^{ot} é dado por:

$$Q_2^{ot} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Descreve-se, a seguir, os resultados de simulação obtidos com o mapeamento proposto.

Resultados de Simulação

A rede de Hopfield modificada foi aplicada na solução do problema das N-rainhas com $N=5$ e com $N=8$. Para $N=5$, a rede converge em cerca de 15 iterações. A Figura 5.3, mostra a representação gráfica para 2 soluções ótimas obtidas pela rede. Para este caso, o número total

de soluções diferentes é igual a 8 (Apêndice D). No problema das N-rainhas com N=8 (dimensão de um tabuleiro oficial), a rede proposta sempre converge para uma solução válida (ponto de equilíbrio) em cerca de 40 iterações em cada simulação. A Tabela 5.2, mostra dez soluções diferentes obtidas pela rede. Deve-se ressaltar que neste caso, para N=8, o número total de soluções diferentes é igual a 92. Todas estas soluções foram obtidas pela rede através da execução de outras simulações (Apêndice D). O vetor de saída da rede, v , foi inicializado, em todas as simulações (independente do valor atribuído a N), com valores aleatórios entre zero e um.

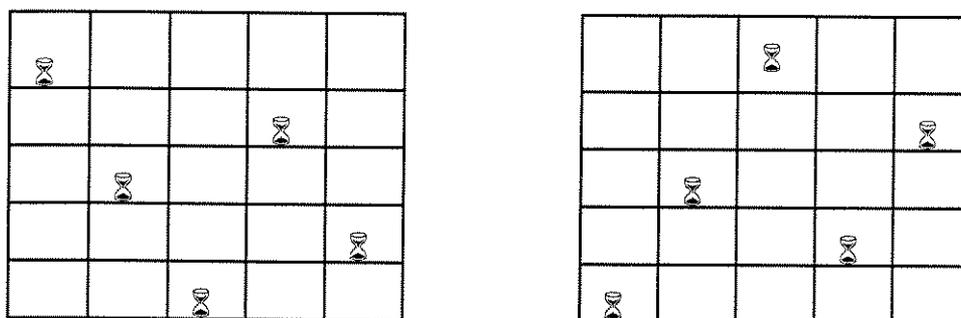


Figura 5.3. Soluções do Problema das N-Rainhas (N = 5)

p	p₁	p₂	p₃	p₄	p₅	p₆	p₇	p₈
1	5	3	1	7	2	8	6	4
2	7	4	2	5	8	1	3	6
3	3	6	2	7	1	4	8	5
4	6	4	7	1	8	2	5	3
5	5	1	8	4	2	7	3	6
6	3	5	2	8	1	7	4	6
7	4	6	8	2	7	1	3	5
8	4	1	5	8	2	7	3	6
9	7	4	2	8	6	1	3	5
10	3	6	8	1	4	7	5	2

Tabela 5.2. Soluções Para o Problema das N-Rainhas (N=8)

A Figura 5.4 mostra, em 3 dimensões, a evolução da matriz V (representação matricial do vetor de saída da rede v , onde $vec(V^T) = v$) numa simulação da rede de Hopfield modificada para o problema das N-rainhas com N=5. Nota-se, nesta figura, que a rede começa a convergir para a solução final, $p^T = [3, 5, 2, 4, 1]$, a partir da quinta iteração. Para o

problema das N-rainhas, as funções de ativação de todos os neurônios são limitadas no intervalo de zero a um.

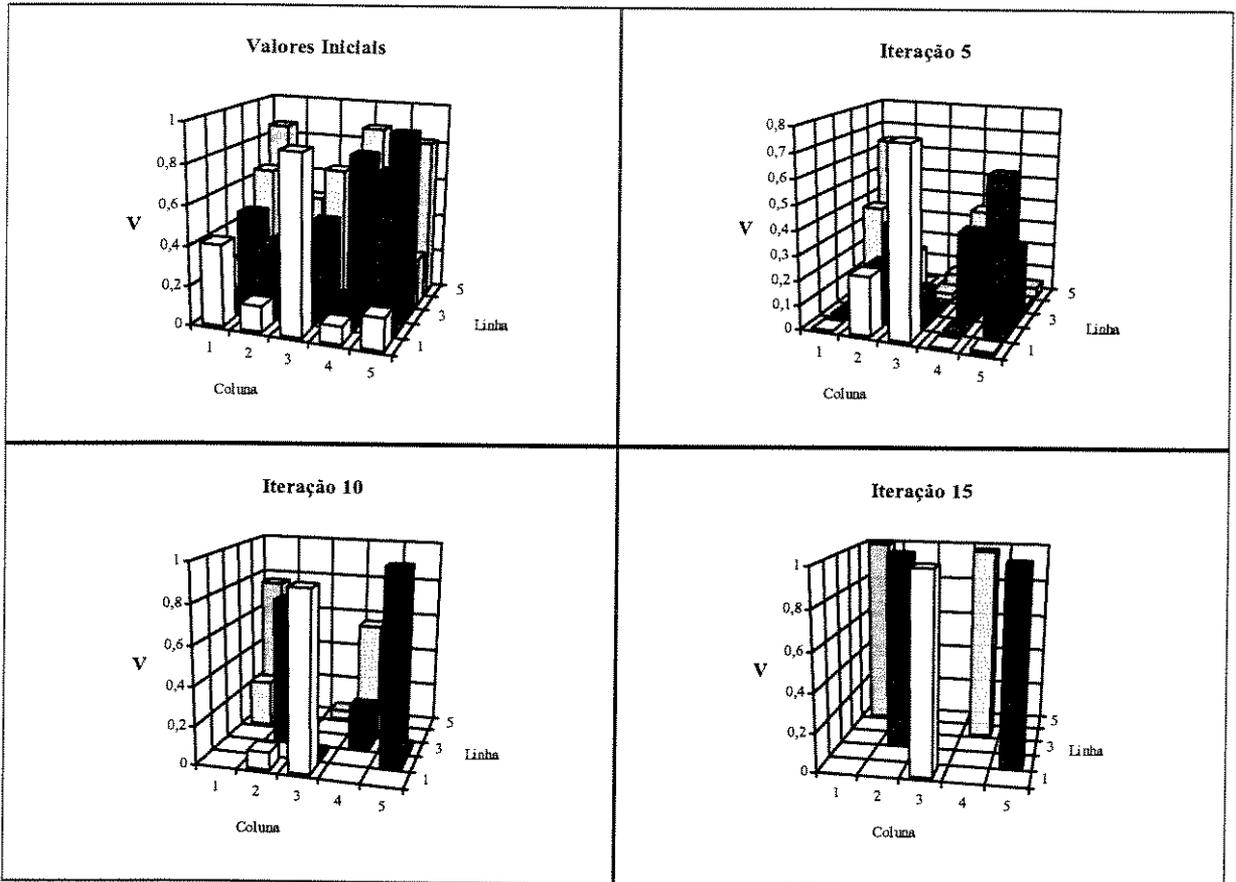
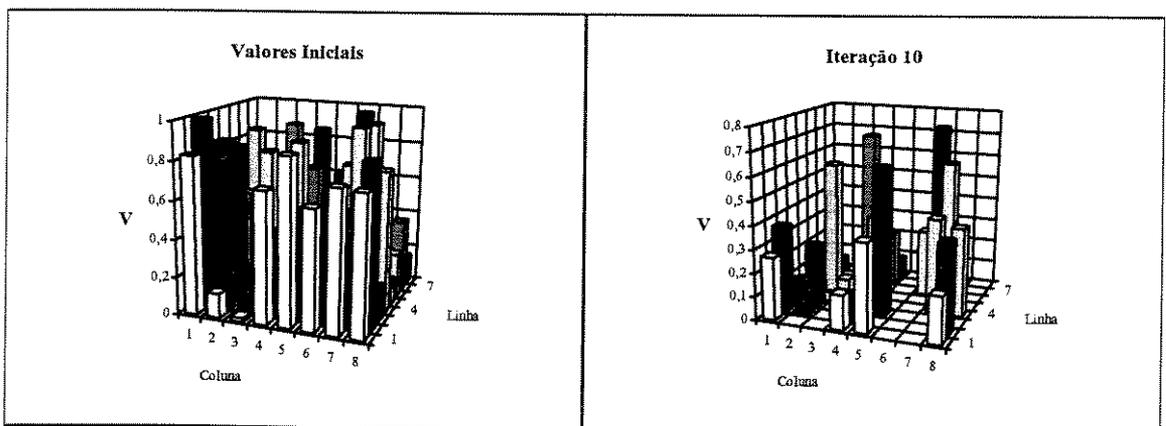


Figura 5.4. Evolução da Matriz V Para o Problema das N-Rainhas Com N=5.

A Figura 5.5 apresenta a evolução da matriz V para N=8. Observa-se que a rede, a partir da décima iteração, já tende para a provável solução final representada pelo vetor $\mathbf{p}^T = [4, 1, 5, 8, 2, 7, 3, 6]$.



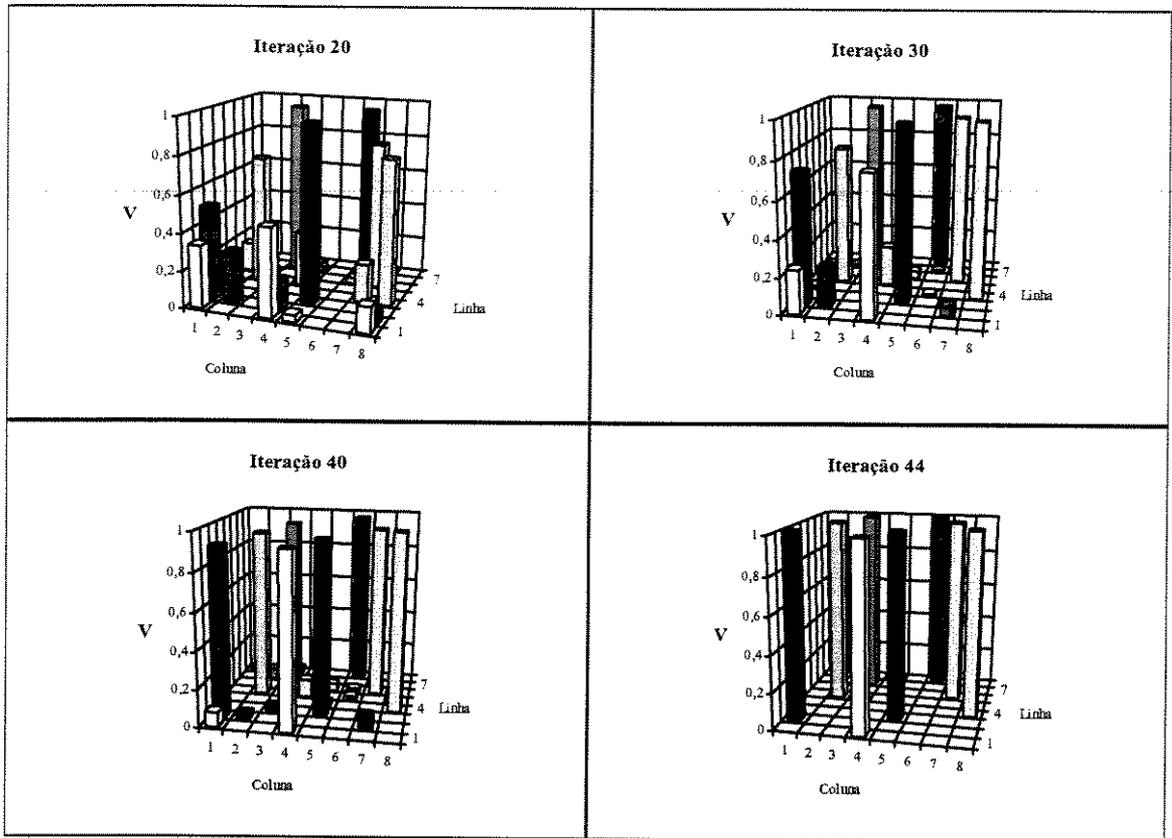


Figura 5.5. Evolução da Matriz V Para o Problema das N-Rainhas Com N=8.

Deve-se ressaltar que, na rede de Hopfield proposta neste trabalho, não há necessidade de especificar nenhuma constante de ponderação. Além disso, os termos de restrições (agrupados em um único termo) e o termo de otimização são tratados em estágios distintos, evitando a interferência entre eles.

5.3.3-) Mapeamento do Problema do Matching-Bipartido

O problema do matching-bipartido está relacionado intrinsecamente à teoria de grafos. Um grafo G é constituído por um par $G = (V,E)$, onde V é um conjunto finito de nodos ou vértices, e o conjunto E tem como elementos os subconjuntos de V , denominado arcos. Para um grafo com quatro vértices (v_1, v_2, v_3 e v_4), tem-se:

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{[v_1,v_2], [v_1,v_3], [v_1,v_4], [v_2,v_3], [v_2,v_4], [v_3,v_4]\}$$

O grafo $G=(V,E)$ é denominado bipartido quando o conjunto de vértices V pode ser particionado em dois conjuntos (com o mesmo número de elementos), U e W , de modo que todos os seus arcos possuem um vértice em U e outro vértice em W . A cada arco $[u_i, w_j]$ associa-se um valor $P_{ij} \geq 0$, correspondendo ao valor de conexão entre o vértice u_i e w_j .

O problema do matching-bipartido em um grafo consiste em determinar um subconjunto M de arcos, de modo que cada vértice de U e W esteja associado somente a um arco. O problema do matching-bipartido é mínimo se a soma dos valores P_{ij} associado a cada arco de M for o menor possível [Papa82]. Como ilustração, a Figura 5.6 mostra um grafo bipartido com quatro vértices. Os conjuntos V , E , U e W , e a matriz \mathbf{P} (escolhida aleatoriamente) são dados por:

$$V = \{u_1, u_2, w_1, w_2\}$$

$$E = \{[u_1, w_1], [u_1, w_2], [u_2, w_1], [u_2, w_2]\}$$

$$U = \{u_1, u_2\}$$

$$W = \{w_1, w_2\}$$

$$\mathbf{P} = \begin{bmatrix} 2.9 & 0.8 \\ 1.3 & 3.5 \end{bmatrix}$$

Neste caso, o matching-bipartido mínimo será escolhido entre o subconjunto $M_1 = \{[u_1, w_1], [u_2, w_2]\}$ e $M_2 = \{[u_1, w_2], [u_2, w_1]\}$. Como a soma dos arcos do subconjunto M_2 é menor que a soma dos arcos do subconjunto definido por M_1 , então os elementos (arcos) do subconjunto M_2 corresponderão ao matching-bipartido mínimo, ou seja, $M = M_2$.

Vários problemas podem ser modelados como ‘problema do matching-bipartido’. Entre estes incluem os sistemas de reconhecimento de objetos em visão computacional [Duda82], problemas envolvendo transmissão de sinais [Ball87], problemas de otimização e sequenciamento de processos operacionais [Papa82], projeto de layout de circuitos impressos [Sind66], e sistemas para desenho de grafos automaticamente [Tama88].

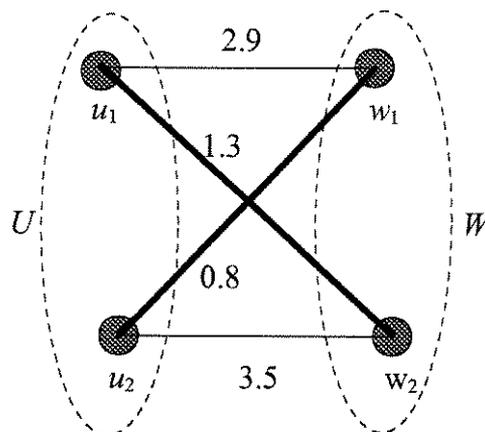


Figura 5.6. Ilustração de Um Grafo Bipartido

Nesta seção, um novo método, utilizando a rede de Hopfield modificada, é desenvolvido para resolver o problema do matching-bipartido. A implementação deste problema, como definido anteriormente, pode ser feito a partir das definições apresentadas na Seção 5.2. Sabe-se que a cada elemento pertencente tanto ao conjunto U como ao conjunto W deve-se associar um único arco. Portanto as restrições estruturais para o problema são semelhantes àquelas do problema das N-rainhas, isto é, cada vértice do conjunto U e W deve ser associado a somente um arco. Redefine-se então o vetor \mathbf{p} como sendo o vetor cujos elementos contêm os índices dos vértices pertencentes ao conjunto W . Assim, o termo p_i representa o arco que une o i -ésimo vértice pertencente a U ao respectivo vértice (valor do elemento p_i) pertencente a W , ou seja:

$$p_i \in \{1, \dots, N\} \text{ onde } i \in \{1..N\}$$

onde 'N' é o número de elementos do conjunto U ou W . Para o problema do matching-bipartido ilustrado na Figura 5.6 com $N=2$, os vetores \mathbf{p} e $\mathbf{v}(\mathbf{p})$, e a matriz $\mathbf{V}(\mathbf{p})$ são dados por:

$$\mathbf{p}^T = [2 \quad 1]$$

$$\mathbf{v}(\mathbf{p})^T = \begin{bmatrix} 0 & 1 & 1 & 0 \\ \delta(p_1)^T & & \delta(p_2)^T & \end{bmatrix}$$

$$\mathbf{V}(\mathbf{p}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Deduz-se a seguir, as equações para os parâmetros internos (\mathbf{T}^{val} , \mathbf{s} , \mathbf{T}^{ot} e \mathbf{i}^{ot}) da rede de Hopfield modificada para o problema do matching-bipartido.

Dedução das Equações Para \mathbf{T}^{val} e \mathbf{s}

As equações de \mathbf{T}^{val} e \mathbf{s} do subespaço-válido para o problema do matching-bipartido são idênticas àquelas desenvolvidas, na Seção 5.3.2, para o problema das N-rainhas. Assim, as equações de \mathbf{T}^{val} e \mathbf{s} do subespaço-válido são dadas por:

$$\mathbf{T}^{\text{val}} = (\mathbf{R}^N \otimes \mathbf{R}^N) \tag{5.54}$$

$$\mathbf{s} = \frac{1}{N} (\mathbf{o}^N \otimes \mathbf{o}^N) \tag{5.55}$$

Portanto, as equações (5.54) e (5.55) garantem que para cada vértice, $u_j \in U$ e $w_i \in W$, seja associado um único arco.

Dedução das Equações Para T^{ot} e i^{ot}

O termo de energia E^{ot} da rede de Hopfield modificada para o problema do matching-bipartido é projetado com o objetivo de que a solução ótima corresponda a menor soma dos valores P_{ij} associado a cada arco pertencente a M . Assim, a soma total (st) referente ao vetor ordenado \mathbf{p} é dada por:

$$st(\mathbf{p}) = \text{traço}(\mathbf{V}(\mathbf{p})^T \cdot \mathbf{P}) \tag{5.56}$$

As equações referentes a T^{ot} e i^{ot} , correspondentes ao termo de energia E^{ot} , podem ser obtidas fazendo-se $E^{ot} = st(\mathbf{p})$, isto é:

$$\begin{aligned} E^{ot} &= st(\mathbf{p}) \\ &= \text{traço}(\mathbf{V}(\mathbf{p})^T \cdot \mathbf{P}) \\ &= \text{vec}(\mathbf{V}(\mathbf{p})^T) \cdot \text{vec}(\mathbf{P}) \\ &= \mathbf{v}(\mathbf{p})^T \cdot \text{vec}(\mathbf{P}) \end{aligned} \tag{5.57}$$

Comparando-se a equação (5.57) com a equação do termo E^{ot} , dada em (2.6), obtêm-se os parâmetros T^{ot} e i^{ot} definidos por:

$$T^{ot} = \mathbf{0} \tag{5.58}$$

$$i^{ot} = -\text{vec}(\mathbf{P}) \tag{5.59}$$

Descreve-se a seguir, os resultados de simulação obtidos pela rede proposta.

Resultados de Simulação

A rede de Hopfield modificada foi utilizada na solução do problema (proposto em [Papa82]) do matching-bipartido com $N=5$. A matriz \mathbf{P} fornecida para o problema, e cujos elementos representam o valor das conexões entre os arcos do grafo bipartido, é dada por:

$$\mathbf{P} = \begin{bmatrix} 7 & 2 & 1 & 9 & 4 \\ 9 & 6 & 9 & 5 & 5 \\ 3 & 8 & 3 & 1 & 8 \\ 7 & 9 & 4 & 2 & 2 \\ 8 & 4 & 7 & 4 & 8 \end{bmatrix}$$

Para as simulações realizadas, a rede converge para a solução ótima, em cerca de 50 iterações. Os valores de \mathbf{p} , $\mathbf{v}(\mathbf{p})$ e $\mathbf{V}(\mathbf{p})$, representando a solução ótima obtida após a convergência da rede, são dados por:

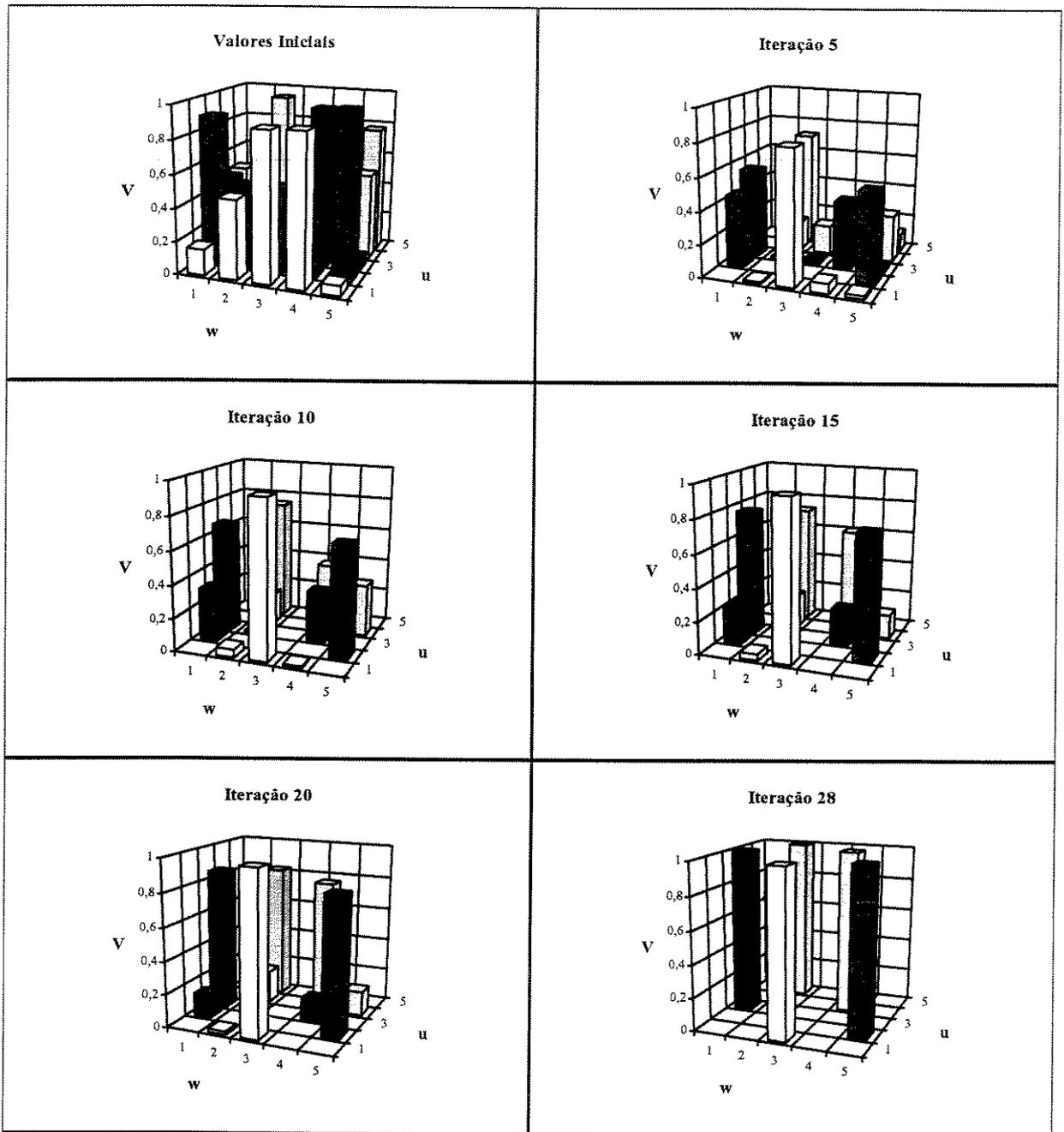


Figura 5.8. Evolução da Matriz V Para o Problema do Matching-Bipartido

5.4-) PROBLEMAS DE PROGRAMAÇÃO DINÂMICA

Programação Dinâmica é um método de otimização utilizado para processos de decisões sequenciais. O termo ‘processo de decisão sequencial’ é utilizado para descrever uma atividade que necessita de uma sequência pré-estabelecida de ações com o propósito de alcançar um objetivo específico.

Assim, programação dinâmica ou programação por estágios é uma técnica muito empregada em problemas que envolvem a otimização de sistemas modelados por uma sequência de estados [Hill80]. Problemas práticos de engenharia elétrica, engenharia aeroespacial, economia e pesquisa operacional são alguns dos problemas que podem ser resolvidos através de programação dinâmica [Lars67]. Através dela, obtém-se um procedimento sistemático para determinar a combinação de decisões que otimize um critério relacionado ao problema.

Para resolver um problema por programação dinâmica é necessário a definição dos estágios e dos estados. As principais propriedades envolvidas com o conceito de estado podem ser definidas por [Dena82]:

- As transições ocorrem de estado para estado.
- Cada estado contém um resumo da história passada do processo, tendo informações suficientes para avaliação das alternativas correntes.

Em cada instante de decisão está associado um estágio onde existem vários estados. Na transição de um estágio para outro, obtém-se, no estágio seguinte, um estado otimizando-se um critério relacionado ao problema em análise. Portanto, a programação dinâmica é um procedimento de otimização, que se aplica a problemas que requerem uma sequência de decisões relacionadas entre si. No procedimento de otimização, procura-se uma sequência de decisões, que por sua vez produz uma sequência de situações que maximiza (ou minimiza) algum objetivo. Normalmente, a resolução de problemas variados através de programação dinâmica, envolve a utilização das relações dinâmicas de recorrência desenvolvidas principalmente por Richard Bellman [Bell57]. Embora a Programação Dinâmica seja uma técnica poderosa para resolver uma variedade muito grande de problemas de otimização, sua utilização tem sido muito limitada devido a relativa ineficiência computacional (e.g. tempo computacional, memória) exigida pelo algoritmo tradicional [Lars67].

A resolução de problemas de programação dinâmica através de redes neurais artificiais foi proposta também em [Chiu91]. Uma breve descrição desta rede é apresentada no Apêndice C. Nesta rede, os termos de otimização e de restrições são tratados em um único estágio. Conseqüentemente, torna-se necessário especificar os parâmetros de ponderação associados a

cada termo de restrição. Para contornar estes problemas, aplica-se a rede de Hopfield modificada na solução de problemas de programação dinâmica.

Nesta seção, faz-se o mapeamento do tradicional problema de programação dinâmica conhecido como ‘problema do caminho mínimo’ utilizando a rede de Hopfield modificada. Um subespaço-válido de soluções é desenvolvido para agrupar todas as restrições estruturais impostas pelo problema.

5.4.1-) Mapeamento de Problemas de Programação Dinâmica Através da Rede de Hopfield Modificada

Um problema de programação dinâmica típico pode ser modelado como um conjunto constituído de um nó fonte e um nó destino com n-estágios intermediários, e m-estados em cada estágio como ilustrado na Figura 5.9, para $n = m = 3$. O custo associado da transição do i-ésimo estado pertencente ao x-ésimo estágio para o j-ésimo estado pertencente ao estágio seguinte, ‘x+1’, é dado pela variável constituída de índices duplos denotada por $d_{xi, (x+1)j}$, onde ‘x’ é o índice dos estágios, e ‘i’ e ‘j’ são os índices dos estados em cada estágio. Para a Figura 5.9, o custo ‘d’ é denotado por $d_{11,22}$. O objetivo é encontrar um caminho composto de um e somente um estado em cada estágio, partindo da fonte com direção ao destino (abordagem forward), e que satisfaça algum critério.

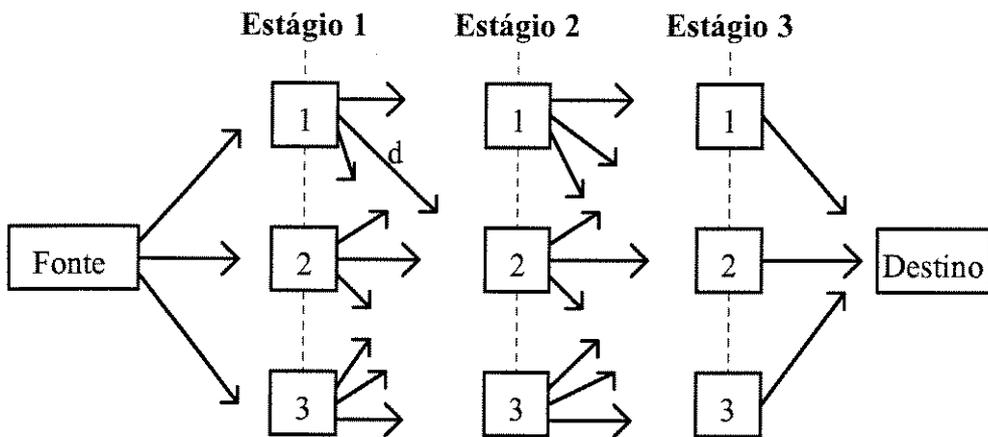


Figura 5.9. O Problema de Programação Dinâmica

Portanto, o objetivo do problema de programação dinâmica considerado nesta seção (problema do caminho mínimo) é encontrar o menor caminho entre todos os caminhos

possíveis, que se inicializa no nó fonte e finaliza no nó destino, passando-se por um único estado em cada estágio. Admite-se que cada nó de estado possa ser convenientemente considerado como um elemento processador do i -ésimo estado pertencente ao x -ésimo estágio.

A implementação do problema de programação dinâmica, como definida anteriormente, pode ser feita a partir das definições apresentadas na Seção 5.2. Deste modo, redefine-se o vetor \mathbf{p} como sendo o vetor que contém os índices dos estados nos diversos estágios intermediários. O valor do elemento p_i representa qual estado está ativado no i -ésimo estágio intermediário, ou seja:

$$p_i \in \{1, \dots, m\} \text{ onde } i \in \{1 \dots n\}$$

sendo ‘ m ’ o número de estados em cada estágio, e ‘ n ’ o número total de estágios intermediários. O vetor $\mathbf{v}(\mathbf{p})$ e a matriz $\mathbf{V}(\mathbf{p})$ são dados por:

$$\mathbf{v}(\mathbf{p}) = \begin{bmatrix} \delta(p_1) \\ \delta(p_2) \\ \delta(p_3) \\ \vdots \\ \delta(p_n) \end{bmatrix}_{n \times m} \qquad \mathbf{V}(\mathbf{p}) = \begin{bmatrix} \delta(p_1)^T \\ \delta(p_2)^T \\ \delta(p_3)^T \\ \vdots \\ \delta(p_n)^T \end{bmatrix}_{n \times m}$$

onde $\delta(k) \in \mathfrak{R}^n$ é um vetor coluna correspondente a k -ésima coluna de δ .

Como exemplo ilustrativo, para um problema de programação dinâmica com 5 estágios e com 4 estados por estágios ($n = 5$ e $m = 4$), e tendo a sequência de estados seguintes como solução: estado 3 no estágio 1, estado 2 no estágio 2, estado 4 no estágio 3, estado 4 no estágio 4, e estado 1 no estágio 5; os vetores \mathbf{p} e $\mathbf{v}(\mathbf{p})$, e a matriz $\mathbf{V}(\mathbf{p})$ são dados por:

$$\mathbf{p}^T = [3, 2, 4, 4, 1]$$

$$\mathbf{v}(\mathbf{p})^T = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\qquad \qquad \qquad \begin{matrix} \delta(p_1)^T & \delta(p_2)^T & \delta(p_3)^T & \delta(p_4)^T & \delta(p_5)^T \end{matrix}$$

$$\mathbf{V}(\mathbf{p}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Deduz-se a seguir, as equações para os parâmetros internos da rede de Hopfield modificada.

Dedução das Equações Para T^{val} e s

As equações de T^{val} e s do subespaço-válido para o problema de programação dinâmica são desenvolvidas utilizando os conceitos descritos na Seção 5.2. A restrição estrutural para o problema de programação dinâmica mostra que apenas um neurônio por estágio deve estar ativado. Esta restrição significa que apenas um e somente um elemento de cada linha da matriz $V(p)$ deve possuir valor igual a um, ou seja:

$$[V(p)]_{ij} \in \{1, 0\}$$

$$\sum_{j=1}^m [V(p)]_{ij} = 1 \tag{5.60}$$

Deve-se ressaltar que não há nenhuma restrição para as colunas da matriz $V(p)$, visto que os índices que representam os estados podem ser repetidos em vários estágios. Assim, como o vetor soma das linhas é dado por $I = V \cdot \mathbf{1}^m$, um subespaço $S \in \mathbb{R}^{n \times m}$ para o problema de programação dinâmica pode ser representado por:

$$S = V = \frac{1}{m} \mathbf{1}^n \cdot \mathbf{1}^{mT} \tag{5.61}$$

A equação (5.61) garante que a soma dos elementos de cada linha da matriz V possui valor igual a um. Portanto, o termo $T^{val} \cdot V$ da equação do subespaço-válido deve garantir que a soma dos elementos de cada linha da matriz V seja igual a zero. Este procedimento pode ser realizado com a utilização da matriz R^m , isto é, a pós-multiplicação de V por R^m transforma a soma das linhas de V para zero, ou seja:

$$V \cdot R^m = T^{val} \cdot V$$

$$I^n \cdot V \cdot R^m = T^{val} \cdot V \tag{5.62}$$

Substituindo-se as equações (5.61) e (5.62) na equação do subespaço-válido, tem-se:

$$V = I^n \cdot V \cdot R^m + \frac{1}{m} \mathbf{1}^n \cdot \mathbf{1}^{mT} \tag{5.63}$$

Aplicando-se em (5.63) o operador $vec(\cdot)$ definido em (5.14), tem-se:

$$\text{vec}(\mathbf{V}) = \text{vec}(\mathbf{I}^n \cdot \mathbf{V} \cdot \mathbf{R}^m) + \frac{1}{m} \text{vec}(\mathbf{o}^n \cdot \mathbf{o}^{mT})$$

$$\text{vec}(\mathbf{V}) = (\mathbf{I}^n \otimes \mathbf{R}^m) \cdot \text{vec}(\mathbf{V}) + \frac{1}{m} (\mathbf{o}^n \otimes \mathbf{o}^m) \tag{5.64}$$

Substituindo-se $\text{vec}(\mathbf{V})$ por \mathbf{v} na equação (5.64), tem-se:

$$\mathbf{v} = (\mathbf{I}^n \otimes \mathbf{R}^m) \cdot \mathbf{v} + \frac{1}{m} (\mathbf{o}^n \otimes \mathbf{o}^m) \tag{5.65}$$

Assim, os parâmetros \mathbf{T}^{val} e \mathbf{s} do subespaço-válido para o problema de programação dinâmica são dados por:

$$\mathbf{T}^{\text{val}} = (\mathbf{I}^n \otimes \mathbf{R}^m) \tag{5.66}$$

$$\mathbf{s} = \frac{1}{m} (\mathbf{o}^n \otimes \mathbf{o}^m) \tag{5.67}$$

destacando-se que as equações (5.66) e (5.67) obedecem as propriedades do subespaço-válido, isto é, $\mathbf{T}^{\text{val}} \cdot \mathbf{T}^{\text{val}} = \mathbf{T}^{\text{val}}$ e $\mathbf{T}^{\text{val}} \cdot \mathbf{s} = \mathbf{0}$.

Dedução das Equações Para \mathbf{T}^{ot} e \mathbf{i}^{ot}

O termo de energia E^{ot} da rede de Hopfield modificada para o problema de programação dinâmica é projetado com o objetivo de que a solução ótima corresponda ao menor percurso entre o nó fonte e o nó destino. Quando o termo de energia E^{ot} for minimizado, a solução ótima desta função de energia corresponderá ao estado de menor energia da rede. A função E^{ot} para o problema de programação dinâmica é definida por:

$$E^{\text{ot}} = \frac{1}{4} \left[\sum_{x=1}^{n-1} \sum_{i=1}^m \sum_{j=1}^m \underbrace{d_{xi,(x+1)j} \cdot v_{xi} \cdot v_{(x+1)j}}_{1\text{o. Termo}} + \sum_{x=2}^n \sum_{i=1}^m \sum_{j=1}^m \underbrace{d_{(x-1)j,xi} \cdot v_{xi} \cdot v_{(x-1)j}}_{2\text{o. Termo}} \right] + \left[\sum_{x=1}^1 \sum_{i=1}^m \underbrace{d_{\text{fonte},xi} \cdot v_{xi}}_{3\text{o. Termo}} + \sum_{x=n}^n \sum_{i=1}^m \underbrace{d_{xi,\text{destino}} \cdot v_{xi}}_{4\text{o. Termo}} \right] \tag{5.68}$$

Nesta equação, o primeiro termo fornece o custo de conexão entre cada neurônio v_{xi} , pertencente ao estágio ‘x’, e todos os outros neurônios que estão no estágio posterior ‘x+1’; enquanto o segundo termo fornece o custo de conexão de cada neurônio v_{xi} com todos os outros neurônios que estão no estágio anterior ‘x-1’. O terceiro termo fornece o custo de conexão entre o nó fonte e todos os outros neurônios que estão no primeiro estágio, enquanto o quarto termo fornece o custo de conexão do nó destino com todos os outros neurônios que

estão no último estágio. Logo, a otimização de E^{ot} corresponde a minimização de cada termo em relação a cada neurônio v_{xi} . Assim, derivando-se a função de energia E^{ot} , dada em (5.68), em relação a v_{xi} (gradiente de E^{ot}), tem-se:

$$-\frac{\partial E^{ot}}{\partial v_{xi}} = -\frac{1}{2} \left[\sum_{j=1}^m \underbrace{d_{xi,(x+1)j} \cdot v_{(x+1)j}}_{x < n} + \sum_{j=1}^m \underbrace{d_{(x-1)j,xi} \cdot v_{(x-1)j}}_{x > 1} \right] - \underbrace{d_{fonte,xi}}_{x=1} - \underbrace{d_{xi,destino}}_{x=n} \quad (5.69)$$

Para obter a matriz \mathbf{T}^{ot} e o vetor \mathbf{i}^{ot} , define-se a matriz $P_{xi,yj}$ como sendo a matriz cujos elementos correspondam ao custo métrico de conexão ligando o i -ésimo neurônio pertencente ao estágio 'x' ao j -ésimo neurônio pertencente ao estágio 'y', ou seja:

$$[\mathbf{P}]_{xi,yj} = \frac{1}{2} d_{xi,yj} \begin{cases} i, j \in \{1..m\} \\ x, y \in \{1..n\} \end{cases} \quad (5.70)$$

Para o problema de programação dinâmica a matriz \mathbf{Q} , representando o encadeamento entre os estágios, é definida por:

$$[\mathbf{Q}]_{xy} = \delta_{(x+1)y} + \delta_{(x-1)y} \quad (5.71)$$

Como ilustração, para $n = 5$, a matriz \mathbf{Q} é dada por:

$$\mathbf{Q} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

A matriz \mathbf{T}^{ot} e o vetor \mathbf{i}^{ot} , pertencente ao termo de energia E^{ot} , deduzidos a partir da equação equação (5.69), são dados por:

$$[\mathbf{T}^{ot}]_{pq} = -[\mathbf{P}]_{xi,yj} \cdot [\mathbf{Q}]_{xy} \quad (5.72)$$

$$\mathbf{i}^{ot} = - \left[\underbrace{d_{fonte,11} \quad d_{fonte,12} \dots d_{fonte,1m}}_m \quad \underbrace{0 \quad 0 \dots 0 \quad 0}_{m \cdot (n-2)} \quad \underbrace{d_{n1,destino} \quad d_{n2,destino} \dots d_{nm,destino}}_m \right] \quad (5.73)$$

onde: $[\mathbf{T}^{ot}]_{pq} \in \mathcal{R}^{nm \times nm}$ e $\mathbf{i}^{ot} \in \mathcal{R}^{nm}$

$$p = m \cdot (x-1) + i$$

$$q = m \cdot (y-1) + j$$

$$x, y \in \{1..n\}$$

$$i, j \in \{1..m\}$$

Na próxima seção, um problema de caminho mínimo é inicialmente solucionado pela rede de Hopfield proposta. Em seguida, compara-se os resultados obtidos pela RHM, nas simulações de vários problemas, com outra rede também utilizada na solução destes problemas.

5.4.2-) Resultados de Simulação

A rede proposta é utilizada para resolver o problema do caminho mínimo mostrado na Figura 5.10, em que os custos de cada caminho $d_{xi,yj}$ (5.70) são os valores indicados na própria Figura 5.10, e o sentido do tráfego é sempre do nó fonte para o nó destino (da esquerda para a direita).

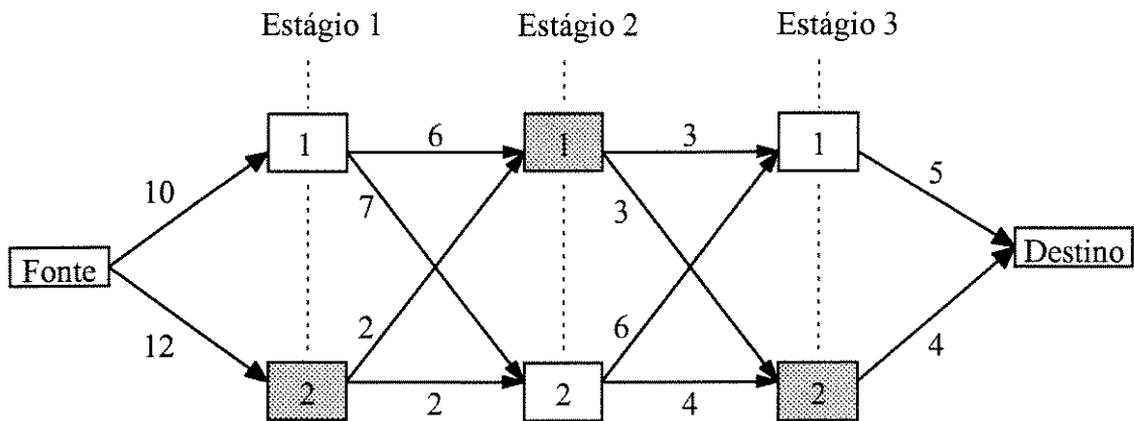


Figura 5.10. Mapa de Caminhos Para o Problema Proposto

A solução ótima para o problema é dada pelos estados que estão hachurados na Figura 5.10, ou seja: estado 2 no estágio 1, estado 1 no estágio 2 e estado 2 no estágio 3. Os vetores \mathbf{p} e $\mathbf{v}(\mathbf{p})$, e a matriz $\mathbf{V}(\mathbf{p})$ representando a solução obtida pela rede são dados por:

$$\mathbf{p}^T = [2 \quad 1 \quad 2]$$

$$\mathbf{v}(\mathbf{p})^T = [0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]$$

$$\mathbf{V}(\mathbf{p}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

A minimização do termo de energia E^{ot} garante que o caminho obtido é o menor entre todos os caminhos possíveis ($E^{ot} = 21$). A rede obteve a solução ótima em somente 3 iterações,

confirmando a sua eficiência. Para o problema do caminho mínimo, a função de ativação de todos os neurônios da rede é limitada no intervalo de 0 a 1.

Para assegurar que a rede de Hopfield modificada pode ser utilizada eficientemente na solução de problemas de programação dinâmica, diferentes problemas foram simulados e comparados com os resultados obtidos pela rede proposta em [Chiu91], que também é baseada na rede de Hopfield convencional.

Número de Estágios (n)	Número de Estados (m)	D ^{RHM}	D ^{RC}
2	2	3.13	3.25
4	4	2.03	3.12
8	8	1.34	2.00
16	16	1.06	1.85
32	32	1.03	1.61
64	64	1.02	1.39
16	2	3.14	3.21
16	4	1.79	2.98
16	8	1.26	1.85
16	32	1.13	1.79
2	16	1.17	1.53
4	16	1.02	1.60
8	16	1.09	1.76

Tabela 5.3. Comparação dos Resultados (Prog. Dinâmica)

Com o propósito de analisar a eficiência da rede de Hopfield modificada, os valores atribuídos aos dados métricos $d_{xi,yj}$ foram selecionados aleatoriamente a partir do conjunto de inteiros composto por {1, 3, 5, 7, 9}. A Tabela 5.3 mostra os resultados obtidos nas simulações. Para ‘n’ e ‘m’ menor que 32, foram executadas 20 simulações diferentes para cada tipo de problema. Para ‘n’ e ‘m’ maior ou igual a 32, foram feitas 10 simulações. Na Tabela 5.3, as colunas ‘D^{RHM}’ e ‘D^{RC}’, mostram respectivamente os resultados do comprimento médio normalizado do caminho obtido pela rede de Hopfield modificada e pela rede proposta em [Chiu91]. O comprimento médio normalizado, D, do caminho é dado por:

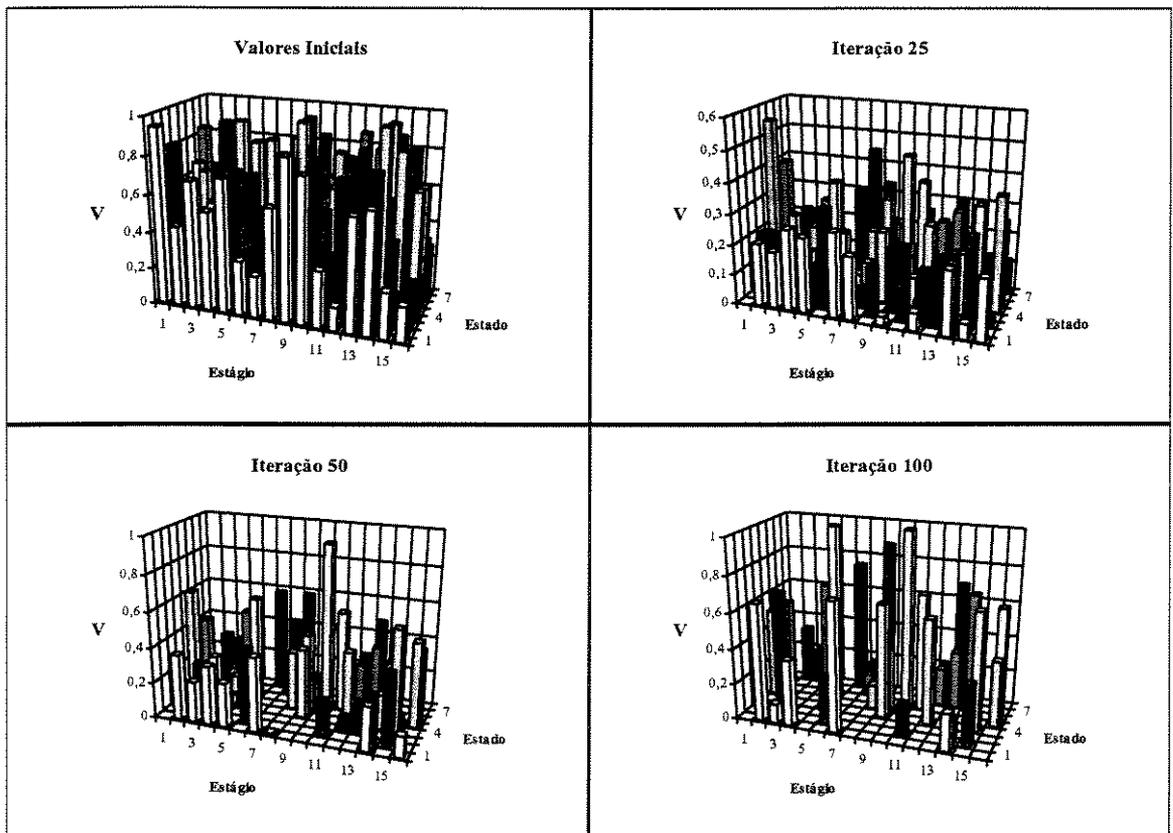
$$D = \frac{S_c}{n_s \cdot (n + 1)} \tag{5.74}$$

onde: S_c é a soma dos caminhos selecionados após a convergência da rede.

n_s é o número de simulações realizadas.

‘n’ é o número de estágios considerado.

Comparando os resultados da Tabela 5.3, nota-se que a rede de Hopfield modificada encontra caminhos menores em relação a rede proposta por [Chiu91]. Este fato deve-se também à dificuldade de se encontrar, para a rede proposta por [Chiu91], os valores adequados para as constantes de ponderações, relativos aos termos de restrições, necessárias para a convergência desta rede. De acordo com as constantes escolhidas, a interferência entre os termos de restrições e o termo de otimização afeta o número de estados mínimos da rede, refletindo-se assim na solução obtida por esta. Outra diferença fundamental entre estas redes, é que a rede proposta em [Chiu91] não utiliza a abordagem do subespaço-válido para confinar as restrições estruturais.



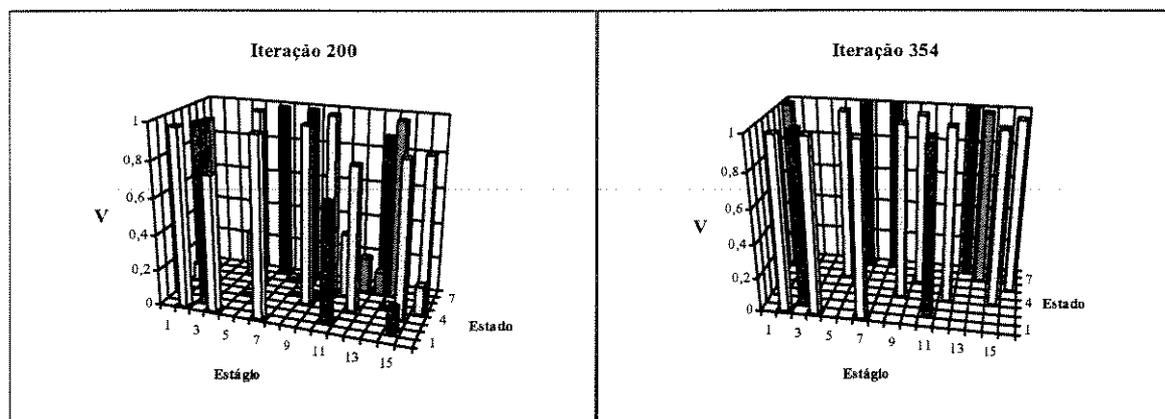


Figura 5.11. Evolução da Matriz V Para o PPD (n=16 e m=8)

A Figura 5.11 mostra, em 3 dimensões, a evolução da matriz V numa simulação da rede de Hopfield modificada para um problema com 16 estágios e 8 estados por estágio. Nota-se que a rede começa a convergir para a solução final, $p^T = [7,1,2,1,6,8,1,8,4,6,2,4,8,7,4,6]$, a partir da centésima iteração.

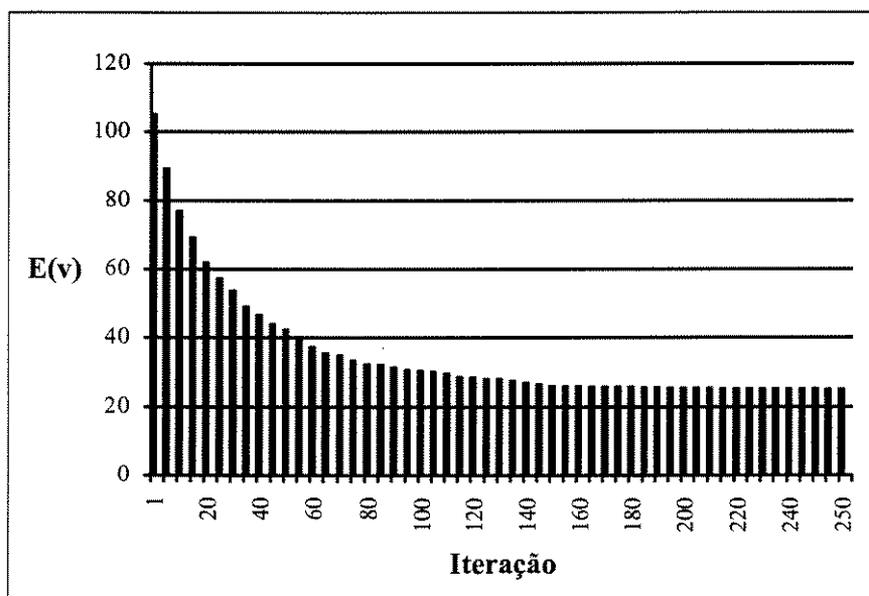


Figura 5.12. Comportamento da Função Objetivo Para o PPD (n=16 e m=8)

A Figura 5.12 ilustra o comportamento da função objetivo do problema $E^{ot}(v)$, representada por $E(v)$, em função do número de iterações. Após a convergência da rede, o valor da função objetivo para o problema (n=16 e m=8) é igual a 25.

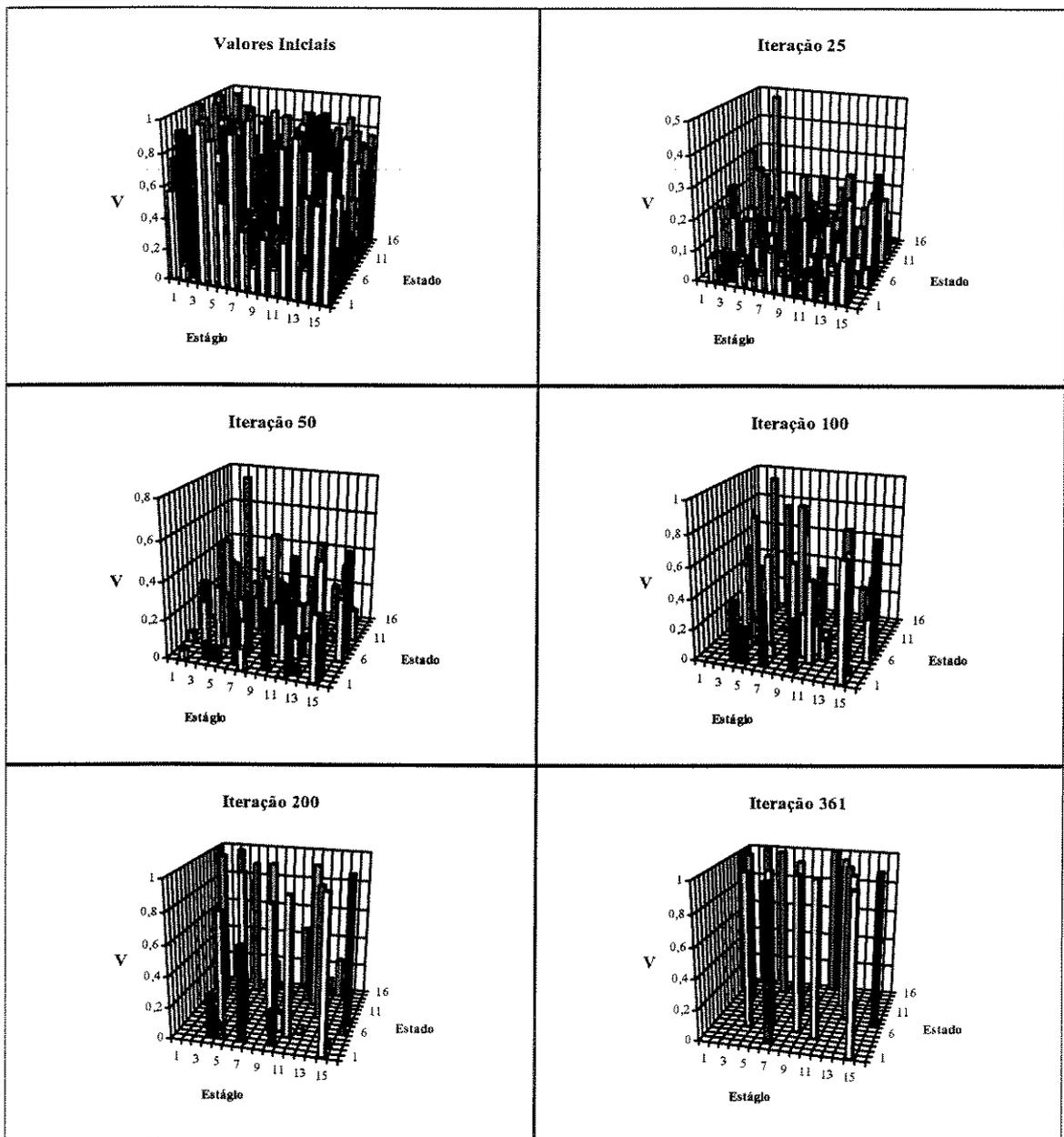


Figura 5.13. Evolução da Matriz V Para o PPD (n=16 e m=16)

A Figura 5.13 mostra a evolução da matriz V numa simulação da rede de Hopfield modificada para um problema com 16 estágios e 16 estados por estágio. A solução final encontrada pela rede é dada por $\mathbf{p}^T = [13,12,15,5,14,6,2,10,5,16,4,12,9,6,1,8]$.

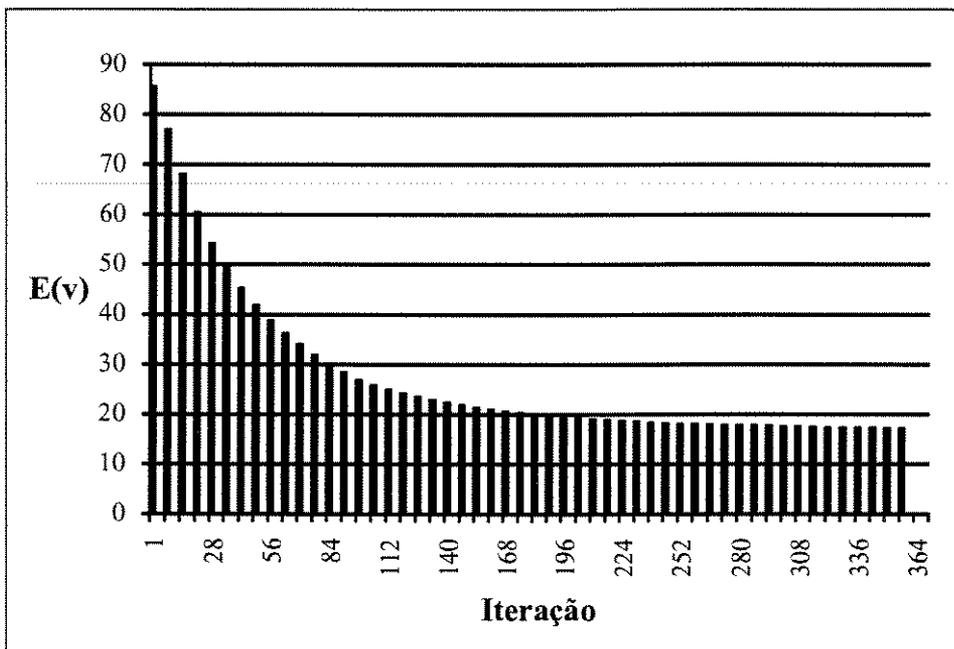


Figura 5.14. Comportamento da Função Objetivo Para o PPD (n=16 e m=16)

A Figura 5.14 ilustra o comportamento da função objetivo em função do número de iterações. Após a convergência da rede, o valor da função objetivo para o problema (n=16 e m=16) é igual a 17.

5.5-) CONCLUSÃO

Neste capítulo, descreveu-se como problemas de otimização combinatorial e problemas de programação dinâmica podem ser solucionados com a rede de Hopfield modificada. Mais especificamente, para problemas de otimização combinatorial, utilizou-se a rede de Hopfield modificada para resolver o problema do caixeiro viajante, o problema das N-rainhas e o problema do matching-bipartido. Vale ressaltar que o mapeamento deste último é inédito. A análise das simulações indica que os resultados obtidos pela rede são semelhantes àqueles obtidos por outras abordagens utilizadas para solucionar problemas de otimização combinatorial.

Em relação aos problemas de programação dinâmica, propõe-se também um novo método para resolver estes problemas através da rede de Hopfield modificada, o qual é aplicado na solução do problema clássico do caminho mínimo. Para este problema, uma função de energia E^{ot} foi projetada com o propósito de fornecer um caminho mínimo entre todos os caminhos possíveis. A fim de validar a metodologia proposta, uma análise comparativa com o

método proposto em [Chiu91], foi realizada através de várias simulações, que comprovaram a eficiência da RHM.

As principais vantagens em utilizar a rede de Hopfield modificada em relação as outras redes neurais, também utilizadas em otimização combinatorial e programação dinâmica, são:

- Não necessidade de especificação de parâmetros de ponderação.
- As restrições estruturais do problema são mapeadas através do subespaço-válido.
- O termo de otimização e termos de restrições da função de energia da rede são tratados em estágios distintos, não havendo interferência entre eles.
- O passo de iteração é calculado dinamicamente através do controlador nebuloso.

Assim, a rede de Hopfield modificada torna-se uma alternativa em relação às redes utilizadas na solução destes problemas.

BIBLIOGRAFIA

[Ball87] D. H. Ball, P. C. Gardner and M. A. Srinivas. *Graph Problems and Connectionist Architectures*. Technical Report, Dep. Comput. Sci., Univ. Rochester, 1987.

[Bell57] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[Chiu91] C. Chinchuan, C. Y. Maa and M. A. Shanblatt. "Energy function analysis of dynamic programming neural networks". *IEEE Trans. on Neural Networks*, vol. 2, no. 4, pp. 418-426, 1991.

[Davi90] Y. Davidor. *An Intuitive Introduction to Genetic Algorithms as Adaptive Optimization Procedures*. Technical Report CS90-07, Department of Applied Mathematics and Computer Science, Weizmann Institute of Science, 1990.

[Dena82] E. V. Denardo. *Dynamic Programming - Models and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1982.

[Duda82] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1982.

[Durb87] R. Durbin and D. Willshaw. "An analogue approach to the travelling salesman problem using an elastic net method". *Nature*, vol. 326, no. 6114, pp. 689-691, 1987.

- [Gare79] M. R. Garey and D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.
- [Glov90] F. Glover. *Tabu search, a tutorial*. Technical Report, Center for Applied Artificial Intelligence, University of Colorado, 1990.
- [Grah81] A. Graham. *Kronecher Products and Matrix Calculus: with Applications*. Ellis Horwood Ltd, Chichester, 1981.
- [Hegd88] S. U. Hegde, J. L. Sweet and W. B. Levy. "Determination of parameters in a Hopfield/Tank computational network". *Proc. of the International Conference on Neural Networks*, San Diego, CA, pp. 291-298, 1988.
- [Hill80] F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research*. Third edition, Holden-Day, San Francisco, CA, 1980.
- [Hopf85] J. J. Hopfield and D. W. Tank. "Neural computation of decisions in optimization problems". *Biological Cybernetics*, vol. 52, pp. 141-152, 1985.
- [Kirk83] S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi. "Optimization by simulated annealing". *Science*, vol. 220, no. 4598, pp. 671-680, 1983.
- [Lars67] R. E. Larson. "A survey of dynamic programming computational procedures". *IEEE Trans. on Automatic Control*, vol. AC-12, pp 767-774, 1967.
- [Mand92] J. Mandziuk and B. Macukow. "A neural network designed to solve the N-queens problem". *Biological Cybernetics*, vol. 66, pp. 375-379, 1992.
- [Papa82] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, Englewood Cliffs, NJ, 1982.
- [Sind66] F. W. Sinden. "Topology of thin film circuits". *Bell Syst. Tech. J.*, vol. XLV, pp. 1639-1666, 1966.
- [Tama88] R. Tamassia, G. Di Battista and C. Batini, "Automatic graph drawing and readability of diagrams". *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-18, pp. 61-79, 1988.
- [Wirt76] N. Wirth. *Algorithm + Data Structures = Programs*. Prentice Hall, Englewood Cliffs, NJ, 1976.

CAPÍTULO 6

IDENTIFICAÇÃO ROBUSTA ATRAVÉS DE REDES NEURAIS ARTIFICIAIS

6.1-) INTRODUÇÃO

A identificação de sistemas está relacionada com o processo de estimar os parâmetros de um modelo, caracterizando o comportamento de um sistema físico, a partir de informações (medidas) obtidas sobre o respectivo sistema. Estas medidas são frequentemente afetadas (corrompidas) pela presença de incertezas (ruídos) que representam algum tipo de perturbação atuando no sistema observado.

Os métodos clássicos de estimação paramétrica, geralmente, consideram que as incertezas associadas às informações coletadas são tipicamente caracterizadas através de um ruído aleatório aditivo, com função densidade de probabilidade (f.d.p.) conhecida. Assim, os estimadores derivados, a partir de um modelo específico, são também variáveis aleatórias; e vários métodos de identificação, baseados na f.d.p. da perturbação e dos dados medidos, podem ser utilizados para calcular a média e a variância dos estimadores [Ljun89, Bill80]. Entretanto, o desempenho destes métodos pode ser degradado se as medidas contiverem aberrações (outliers), rupturas de modelo, ou uma f.d.p. desconhecida [Arru92].

Recentemente, uma abordagem alternativa à caracterização estocástica das incertezas tem sido desenvolvida, onde considera-se que a perturbação atuando no processo é limitada (abordagem UBBE: Unknown-But-Bounded Error Approach) [Clem88, Walt90]. Os métodos desta abordagem, calculam uma região do espaço paramétrico (Região de Pertinência

Paramétrica - RPP), que é compatível com a estrutura do modelo, com os dados medidos e com os limites previamente conhecidos das perturbações. Qualquer vetor de parâmetros pertencente a esta região é considerado um estimador dos parâmetros reais do processo. A caracterização das incertezas para estes estimadores é dada por um intervalo (Intervalo de Incerteza Paramétrica - IIP) de valores possíveis para cada parâmetro estimado. Este intervalo é obtido a partir dos valores extremos assumidos por cada estimador na região de pertinência paramétrica.

A abordagem UBBE tem significado prático em diversas aplicações, como por exemplo, em controle quando a informação disponível sobre a perturbação é fornecida como limites sobre seu valor absoluto, em aplicações onde os dados medidos não são suficientes para garantir resultados assintóticos confiáveis, e para diagnósticos de falhas, onde as perturbações correspondem às flutuações dos valores dos componentes em torno de seus valores nominais [Arru92].

Neste capítulo propõe-se a utilização da rede de Hopfield modificada, proposta na Seção 2.4, para resolver problemas de identificação robusta. Alguns dos resultados apresentados têm sido inicialmente desenvolvidos em [Silv95, Silv97]. Na Seção 6.2 descreve-se como é realizada a formulação destes problemas utilizando a abordagem UBBE. O mapeamento de problemas de identificação paramétrica robusta através da rede de Hopfield modificada é realizado na Seção 6.3. Em seguida, a rede desenvolvida é aplicada na solução de problemas de estimação robusta para modelos lineares e não-lineares. No caso linear, os resultados obtidos são comparados a outros métodos utilizados na solução destes problemas.

6.2-) FORMULAÇÃO DO PROBLEMA

Na abordagem UBBE, o modelo utilizado para representar o comportamento dinâmico do processo, com perturbações limitadas, é descrito por uma relação $F(\cdot)$, linear ou não-linear, que relaciona o vetor de parâmetros reais $\theta^* \in \mathcal{R}^N$, que se deseja estimar, e a saída do processo $y(t)$ perturbado por um ruído aditivo $w(t)$:

$$y(t) = F(\theta^*, \phi(t)) + w(t); \quad t \geq 0 \quad \text{e} \quad \phi(t) \in \mathcal{R}^N \quad (6.1)$$

onde: $\phi(t)$ é um vetor de variáveis conhecidas inerentes ao processo.

$F(\theta^*, \phi(t))$ é uma família de modelos que pode representar os dados medidos.

$w(t)$ é a perturbação atuando no processo.

A perturbação $w(t)$ agindo na saída observada pode incluir erros de modelagem, ruídos de medida, erros de arredondamento e outros tipos de incertezas. Na abordagem UBBE, assume-se que $w(t)$ está limitado, ou seja:

$$|w(t)| \leq \delta(t); \quad \delta(t) \geq 0 \tag{6.2}$$

onde $\delta(t)$ é supostamente conhecido.

Os métodos de identificação robusta sob hipótese de erro limitado, calculam a menor região $M(k, \theta) \in \mathcal{R}^N$, que é compatível com a estrutura do modelo (6.1), com os limites para a perturbação (6.2) e com as medidas $\{y(t), t \in [1..k]\}$. A região $M(k, \theta)$ é definida por:

$$M(k, \theta) \equiv \bigcap_{t=1}^k H(t, \theta) \tag{6.3}$$

$$H(t, \theta) = \{ \theta: (y(t) - F(\theta^*, \phi(t)))^2 \leq (\delta(t))^2 \} \tag{6.4}$$

Para modelos lineares, $H(t, \theta)$ é limitado por dois hiperplanos paralelos, ortogonais ao vetor $\phi(t)$. Assim, a região $M(k, \theta)$ é um poliedro convexo resultante da interseção de ‘2k’ hiperplanos em \mathcal{R}^N . Um exemplo desta região é ilustrado na Figura 6.1 para $n=2$ e $k=5$. A determinação de $M(k, \theta)$ torna-se complicada quando o número de parâmetros estimados e/ou o número de medidas é elevado. Uma solução simplificada consiste em aproximar $M(k, \theta)$ por regiões geométricas mais simples como elipsóides [Foge82] ou hipercubos [Vici89]. Uma síntese destas diferentes soluções pode ser encontrada em [Arru92, Silv95, Walt90].

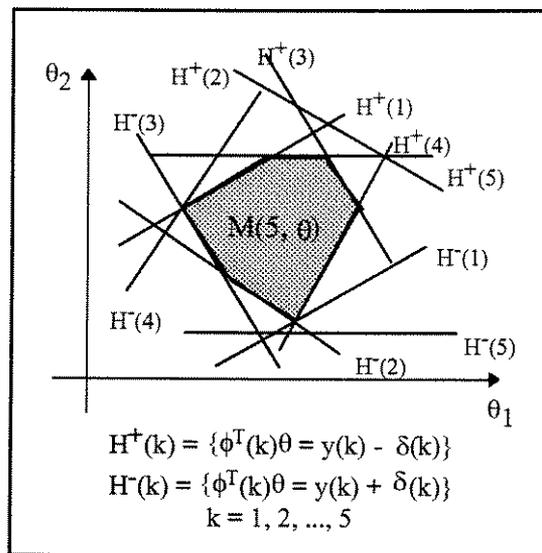


Figura 6.1. A Região de Pertinência Paramétrica

Para modelos não-lineares, $H(t, \theta)$ não é mais limitada por hiperplanos paralelos e sim por hipersuperfícies. Como resultado, a região $M(k, \theta)$ definida pela interseção de $H(t, \theta)$ pode ser não convexa, e/ou constituída por sub-regiões não conectadas. Geralmente, uma aproximação simplificada de $M(k, \theta)$, quando se utilizam modelos não-lineares, é obtida por meio de uma região hipercúbica mínima que seja externa a $M(k, \theta)$ e esteja alinhada com os eixos paramétricos [Belf88].

Em particular, a região hipercúbica mínima contendo a região de pertinência paramétrica, tanto para modelos lineares como para modelos não-lineares, possui as seguintes características:

- As faces da região hipercúbica automaticamente delimitam os intervalos de incerteza paramétrica.
- O centro da região hipercúbica, denominado estimador central, é considerado um estimador dos parâmetros desconhecidos do modelo [Mila85, Kace86].

Os intervalos de incerteza paramétrica IIP_i , correspondente ao i -ésimo parâmetro desconhecido, é definido por:

$$IIP_i = [\theta_i^m, \theta_i^M], \quad i \in \{1..N\} \quad (6.5)$$

onde os parâmetros θ_i^m e θ_i^M são dados por:

$$\theta_i^m = \min_{\theta_i \in M(k, \theta)} \theta_i, \quad i \in \{1..N\} \quad (6.6)$$

$$\theta_i^M = \max_{\theta_i \in M(k, \theta)} \theta_i, \quad i \in \{1..N\} \quad (6.7)$$

Assim, a região hipercúbica mínima (RH^{Min}) contendo a região de pertinência paramétrica é obtida pelo produto cartesiano dos IIP_i , ou seja:

$$RH^{Min} = [IIP_1 \times IIP_2 \times \dots \times IIP_N] \quad (6.8)$$

Finalmente, o estimador central θ^c é calculado por [Mila85]:

$$\theta_i^c = \frac{(\theta_i^m + \theta_i^M)}{2}, \quad i \in \{1..N\} \quad (6.9)$$

Neste capítulo, apresenta-se uma nova abordagem utilizando a rede de Hopfield modificada para resolver problemas de identificação paramétrica robusta com erro desconhecido-mas-limitado. Para modelos lineares, a rede é aplicada para obter tanto os intervalos de incerteza paramétrica como uma aproximação politópica da região de pertinência paramétrica. Para modelos não-lineares, a rede de Hopfield modificada é utilizada para obter os intervalos de incerteza paramétrica, e conseqüentemente o estimador central de θ^* . Na próxima seção, faz-se o mapeamento de problemas de identificação robusta através da rede de Hopfield modificada.

6.3-) MAPEANDO PROBLEMAS DE IDENTIFICAÇÃO ROBUSTA ATRAVÉS DA REDE DE HOPFIELD MODIFICADA

Sendo M^{med} o número de medidas da saída de um processo modelado pelas equações (6.1) e (6.2), o cálculo da região de pertinência paramétrica e dos intervalos de incerteza paramétrica pode ser efetuado a partir do conjunto de restrições dado por (6.4).

Para modelos lineares, a função $F(\theta^*, \phi(t))$ pode ser rescrita como um modelo de regressão linear, ou seja:

$$F(\theta^*, \phi(t)) = \phi(t)^T \cdot \theta^* \tag{6.10}$$

Logo, a equação (6.1) torna-se:

$$y(t) = \phi(t)^T \cdot \theta^* + w(t) \tag{6.11}$$

Substituindo (6.10) em (6.4), tem-se:

$$H(t, \theta) = \{ \theta : y(t) - \delta(t) \leq \phi(t)^T \cdot \theta \leq y(t) + \delta(t) \} \tag{6.12}$$

O cálculo da região de pertinência paramétrica e dos intervalos de incerteza, para modelos lineares, pode ser realizado por uma rede de Hopfield modificada que resolve o seguinte problema de otimização:

$$\text{Minimizar } E^{ot}(\theta) = -\frac{1}{2} \theta^T \cdot T^{ot} \cdot \theta - \theta^T \cdot i^{ot} \tag{6.13}$$

$$\text{sujeito a: } \phi^T \cdot \theta \leq \mathbf{b} \tag{6.14}$$

$$z_i^{min} \leq \theta_i \leq z_i^{max}, \quad i \in \{1..N\} \tag{6.15}$$

onde $\theta \in \mathfrak{R}^N$ é o conjunto de vértices do poliedro $M(k, \theta)$, $\phi \in \mathfrak{R}^{N \times 2M^{med}}$ é uma matriz de variáveis conhecidas inerentes ao processo, \mathbf{z}^{\min} e \mathbf{z}^{\max} são limites que especificam um intervalo de incerteza paramétrica inicial para cada estimador; e $\mathbf{b}^T = [\mathbf{b}^{\min T} \ \mathbf{b}^{\max T}] \in \mathfrak{R}^{2N}$ é um vetor de medidas da saída do processo levando em conta os limites conhecidos de $\delta(t)$, ou seja:

$$\mathbf{b}_k^{\min} = y(k) - \delta(k) \quad , \quad i \in \{1..N\} \quad (6.16)$$

$$\mathbf{b}_k^{\max} = y(k) + \delta(k) \quad , \quad i \in \{(N+1)..2N\} \quad (6.17)$$

As condições (6.14) e (6.15) definem um poliedro convexo limitado, como descrito pela equação (6.3), dentro do qual o vetor θ deve estar contido se este representa um solução válida do problema de otimização quadrática (6.13). Esta solução pode ser obtida com a rede de Hopfield modificada cujo subespaço-válido garante que as condições definidas em (6.14) sejam satisfeitas, isto é, $\theta \in M(k, \theta)$, enquanto a função objetivo E^{ot} é minimizada. O hiper-cubo inicial representado pelas desigualdades dadas em (6.15) é implementado diretamente através da função rampa-simétrica usada como função de ativação da rede.

Dedução das Equações Para \mathbf{T}^{val} , \mathbf{s} , \mathbf{T}^{ot} e \mathbf{i}^{ot} (Modelos Lineares)

Desde que as restrições de desigualdade, dadas por (6.14), para o problema de identificação robusta são idênticas às definidas por (3.3) para o problema de programação linear, os parâmetros \mathbf{T}^{val} e \mathbf{s} do subespaço-válido são os mesmos fornecidos pelas equações (3.16) e (3.17), ou seja:

$$\mathbf{T}^{val} = \mathbf{I} - \phi^+ . (\phi^{+T} . \phi^+)^{-1} . \phi^{+T} \quad (6.18)$$

$$\mathbf{s} = \phi^+ . (\phi^{+T} . \phi^+)^{-1} . \mathbf{b}^+ \quad (6.19)$$

Por outro lado, a minimização de E^{ot} deve conduzir aos vértices da região de pertinência paramétrica e aos valores dos intervalos de incerteza paramétrica. Como para os problemas de programação quadrática, assume-se que a matriz \mathbf{T}^{ot} seja nula. Neste caso, a função E^{ot} depende apenas do vetor \mathbf{i}^{ot} e os vértices de $M(k, \theta)$ podem ser calculados pela seleção de vetores apropriados que serão definidos por duas perspectivas de seleção: perspectiva eixos-direcionados e perspectiva eixos-hipercúbicos [Silv95]. Estas perspectivas alteram o termo de energia E^{ot} com o objetivo de alcançar os pontos de equilíbrio (vértices de $M(k, \theta)$) da rede.

Na perspectiva eixos-direcionados, os vetores \mathbf{i}^{ot} são os vetores unitários que estão na direção direta e inversa a cada eixo paramétrico como ilustrado na Figura 6.2 (a) para $N=2$. Sendo ‘N’ a dimensão do espaço paramétrico (ou quantidade de parâmetros a ser estimado), o número de vetores \mathbf{i}^{ot} a ser selecionado é dado por $NV^{ed} = 2N$. A utilização desta perspectiva, tanto em modelos lineares como não-lineares, leva ao cálculo imediato dos intervalos de incerteza paramétrica para os estimadores. Isto é, a rede de Hopfield modificada que minimiza E^{ot} através de uma perspectiva eixos-direcionados é um método de estimação robusta que aproxima a região de pertinência paramétrica por uma região hipercúbica externa mínima.

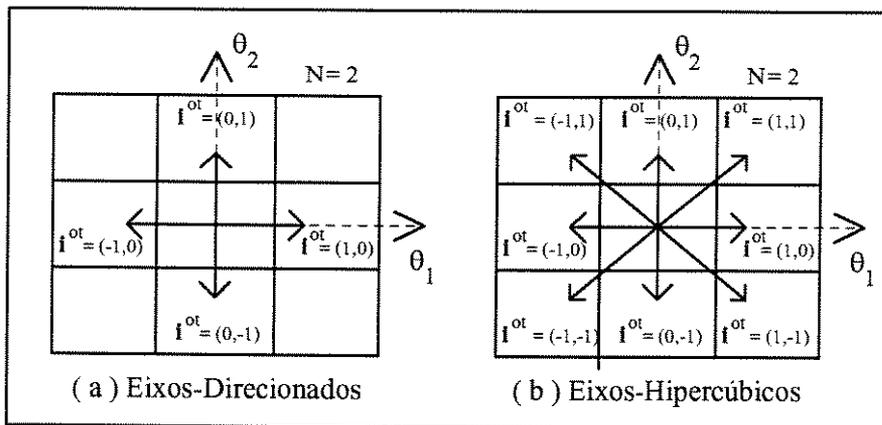


Figura 6.2. Perspectivas de Seleção de \mathbf{i}^{ot}

Na perspectiva eixos-hipercúbicos, os vetores \mathbf{i}^{ot} selecionados são os vetores unitários que ligam o centro de um hipercubo (alinhado com os eixos paramétricos) centrado na origem aos seus próprios vértices e também aos centros de suas faces, como mostrado na Figura 6.2 (b) para $N=2$. O número de vetores escolhidos nesta perspectiva é dado por $NV^{eh} = 2N+2^N$. Como o método proposto em [Mo90], a rede de Hopfield que minimiza E^{ot} , através de uma perspectiva eixos-hipercúbicos, fornece uma descrição politópica (poliédrica) da região de pertinência paramétrica. Neste caso, o centro do poliedro é considerado um estimador válido de θ^* .

Dedução das Equações Para T^{val} , s , T^{ot} e \mathbf{i}^{ot} (Modelos Não-Lineares)

Nos problemas de identificação robusta para modelos não-lineares, utiliza-se uma metodologia semelhante àquela definida pelas equações (6.13) a (6.14), em que as restrições de desigualdade lineares dadas por (6.12) são substituídas por:

$$H(t, \theta) = \{ \theta: y(t) - \delta(t) \leq F(\theta) \leq y(t) + \delta(t) \} \quad (6.20)$$

onde $F(\theta)$ é um modelo não-linear que pode representar os dados medidos.

Logo, a equação definida por (6.14) torna-se:

$$\mathbf{h}(\theta) = \mathbf{F}(\theta) - \mathbf{b} \leq \mathbf{0} \quad (6.21)$$

Desde que a equação (6.21) seja semelhante àquela definida por (4.3) para o problema de otimização não-linear restrito, as equações para \mathbf{T}^{val} e \mathbf{s} , obtidas como na Seção 4.3.1, serão dadas por:

$$\mathbf{T}^{\text{val}} = \mathbf{I} - \nabla \mathbf{h}(\theta)^T \cdot (\nabla \mathbf{h}(\theta) \cdot \nabla \mathbf{h}(\theta)^T)^{-1} \cdot \nabla \mathbf{h}(\theta) \quad (6.22)$$

$$\mathbf{s} = \mathbf{0} \quad (6.23)$$

Para modelos não-lineares, a rede de Hopfield modificada é utilizada no cálculo dos intervalos de incerteza paramétrica e os estimadores centrais de θ^* . Para tanto, utiliza-se a perspectiva eixos-direcionados definida anteriormente; e assume-se, que \mathbf{T}^{of} seja nula. Quando a região de pertinência paramétrica é formada por sub-regiões não conectadas, uma alternativa conveniente é o cálculo dos intervalos de incerteza para cada sub-região [Belf88].

Na próxima seção, apresenta-se alguns exemplos utilizando a rede de Hopfield modificada para solucionar problemas de identificação robusta.

6.4-) SIMULAÇÕES DE PROBLEMAS DE IDENTIFICAÇÃO ROBUSTA

Para ilustrar o desempenho da rede proposta, simulou-se três problemas de identificação paramétrica robusta, sendo dois lineares e um não-linear. Nos problemas envolvendo modelos lineares, os resultados são comparados com aqueles obtidos utilizando métodos clássicos de identificação robusta.

6.4.1-) Identificação Robusta Para Modelos Lineares

Para modelos lineares, a rede foi aplicada na estimação de um modelo ARMAX (Auto Regressive Moving Average with eXternal input) de quarta ordem, e também em um problema prático de obtenção de um modelo para medir a qualidade da água de um lago.

Processo 1 - Estimação de parâmetros de um modelo ARMAX de quarta ordem

Seja o modelo ARMAX de quarta ordem, dado por:

$$y(k) = a_1.y(k-1) + a_2.y(k-2) + b_1.u(k-1) + b_2.u(k-2) + e(k) \equiv \phi(k)^T.\theta + e(k) \quad (6.24)$$

onde: $y(k)$ é o vetor de medidas do processo ($k = 1..50$).

$u(k)$ é uma sequência binária pseudo-aleatória com amplitude ± 1 .

a_1, a_2, b_1, b_2 são os parâmetros desconhecidos do modelo.

$e(k)$ é uma sequência, desconhecida mas limitada, representada por uma distribuição normal $\{N(0,1)\}$ limitada em ± 1 .

Os dados foram gerados a partir de (6.24) com valores nominais para os parâmetros definidos por:

$$a_1 = 1.5, \quad a_2 = -0.7, \quad b_1 = 1.0 \quad e \quad b_2 = 0.5$$

A Tabela 6.1 compara os Intervalos de Incerteza Paramétrica (IIP) calculados pela rede de Hopfield modificada, combinada com a perspectiva eixos-direcionados, com os resultados obtidos utilizando o método ortotopo-externo proposto por [Clem88]. Nesta tabela, a coluna intitulada ‘centro’ representa o estimador central dos parâmetros desconhecidos do modelo.

IIP	RHM			Ortotopo-Externo		
	[min , max]	largura	centro	[min , max]	largura	centro
a_1	[1.3053 , 1.6673]	0.3620	1.4863	[1.3084 , 1.6296]	0.3212	1.4690
a_2	[-0.8200 , -0.5616]	0.2584	-0.6908	[-0.8158 , -0.5782]	0.2376	-0.6970
b_1	[0.6817 , 1.3752]	0.6935	1.0284	[0.7092 , 1.3302]	0.6210	1.0197
b_2	[0.1223 , 0.8911]	0.7688	0.5067	[0.1642 , 0.7669]	0.6027	0.4656

Tabela 6.1. Intervalos de Incerteza Paramétrica (Processo 1)

Na Tabela 6.2, os resultados obtidos pela rede de Hopfield modificada, combinada com a perspectiva eixos-hipercúbicos, são comparados com aqueles obtidos quando se utiliza o método politopo-exato proposto por [Mo90]. Nestes casos, os estimadores calculados a partir do centro de massa de cada região politópica são também considerados estimadores válidos de θ^* . A coluna ‘Erro H-N’, na Tabela 6.2, representa o erro relativo {definido em (3.30)} entre os estimadores obtidos pela rede e os parâmetros nominais do modelo; e a coluna ‘Erro P-N’ é o erro relativo entre os estimadores obtidos pelo método politodo-exato e os parâmetros nominais do modelo.

Estimadores	RHM	Politopo-Exato	Erro H-N	Erro P-N
a_1	1.4811	1.4544	1.260 %	3.040 %
a_2	-0.6929	-0.6967	1.014 %	0.471 %
b_1	1.0298	1.0354	2.980 %	3.540 %
b_2	0.4872	0.4432	2.560 %	11.360 %

Tabela 6.2. Estimadores Paramétricos (Processo 1)

Na Figura 6.3, a linha pontilhada corresponde à saída do modelo utilizando os parâmetros estimados pela rede (Tabela 6.2) em função de um novo conjunto de medidas, enquanto a linha contínua corresponde aos valores da saída do modelo quando se utilizam os parâmetros nominais.

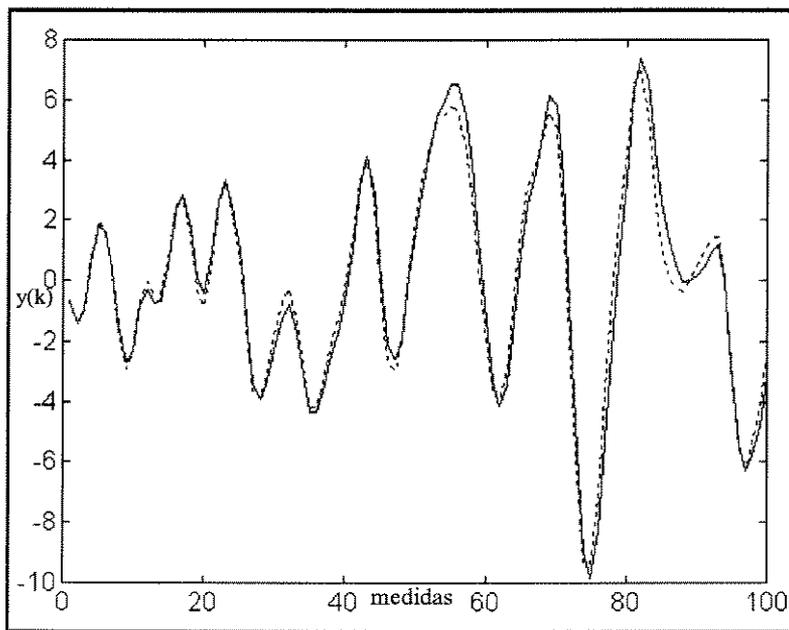


Figura 6.3. Saída do Modelo em Função de Um Conjunto de Medidas

Estes resultados mostram a eficiência da rede de Hopfield modificada na estimação robusta de regiões de pertinência paramétrica para modelos lineares. Dependendo da perspectiva utilizada para a escolha dos vetores i^o , a rede pode operar como o método ortotopo-externo [Clem88] ou como o método politopo-exato [Mo90].

Processo 2 - Estimação de parâmetros de um modelo para qualidade de água

A variação diurna de oxigênio dissolvido em um lago deve ser modelada via estimação de parâmetros de uma equação à diferenças. As informações foram obtidas a partir do lago “De Poel en’t Zwet” situado na região Oeste da Holanda, entre os dias 21 e 30 de Abril de

1983 [Kees87]. A taxa de variação de oxigênio dissolvido no lago em função de diversos fatores (mudança climática, produção de fotossíntese, biodegradação, respiração, etc.) é descrito pela seguinte equação:

$$\dot{c}(t) = K_r[C_s(t) - c(t)] + K_f \cdot I(t) - R \tag{6.25}$$

onde: $c(\cdot)$ é a concentração de oxigênio dissolvido (g/m^3).

$C_s(\cdot)$ é o grau de saturação (g/m^3).

$I(\cdot)$ é a radiação solar (W/m^2).

Os parâmetros desconhecidos do modelo a serem estimados são definidos por:

K_r : coeficiente de aeração (1/d).

K_f : coeficiente do grau de fotossíntese (g/mdW).

R : grau de retrocesso da variação.

Uma forma discreta deste modelo é representada por:

$$c(k) = K_r[C_s(k-1) - c(k-1)] + K_f \cdot I(k-1) - R, \quad k = 1..100 \tag{6.26}$$

$$y(k) = c(k) + e(k) \tag{6.27}$$

onde $y(k)$ é a saída do modelo e o termo $e(k)$ representa a perturbação que é desconhecida mas limitada por $|e(k)| \leq 1,5$. Os dados utilizados foram obtidos a partir de [Kees87], com espaço paramétrico inicial definido por:

$$\Omega = \begin{cases} K_r \in [0.5, 1.8] \\ K_f \in [0.03, 0.07] \\ R \in [0.9, 3.5] \end{cases}$$

O espaço paramétrico inicial Ω é implementado através da função de ativação rampa-simétrica definida em (2.10). Os resultados obtidos são fornecidos pelas tabelas seguintes.

IIP	RHM			Ortotopo-Externo		
	[min , max]	largura	centro	[min , max]	largura	centro
K_r	[0.5676 , 1.7202]	1.1526	1.1439	[0.5685 , 1.7128]	1.1443	1.1407
K_f	[0.0359 , 0.0653]	0.0294	0.0506	[0.0332 , 0.0659]	0.0327	0.0496
R	[1.0213 , 3.4815]	2.4602	2.2514	[0.9923 , 3.4856]	2.4933	2.2390

Tabela 6.3. Intervalos de Incerteza Paramétrica (Processo 2)

A Tabela 6.3 compara os intervalos de incerteza paramétrica obtidos pela rede de Hopfield modificada em relação aos resultados obtidos pelo método ortotopo-externo [Clem88]. A Tabela 6.4 mostra que os estimadores paramétricos obtidos pela RHM e pelo método politopo-exato são bem próximos, validando assim o desempenho de rede proposta.

Estimadores	RHM	Politopo-Exato
K_r	1.2298	1.2349
K_f	0.0537	0.0541
R	1.7513	1.7492

Tabela 6.4. Estimadores Paramétricos (Processo 2)

Para este exemplo, a rede de Hopfield modificada foi utilizada para estimar os parâmetros de um modelo representando um processo dinâmico real. É importante ressaltar que o aumento do número de medidas não degrada o desempenho de rede. Os resultados confirmam a validade e eficiência da rede quando dados experimentais realísticos são utilizados no modelo estimado.

6.4.2-) Identificação Robusta Para Modelos Não-Lineares

Para modelos não-lineares, a rede foi utilizada no cálculo dos intervalos de incerteza paramétrica para o problema de estimação paramétrica robusta definido pelo seguinte modelo:

$$y(k) = \begin{cases} \theta_2 - \theta_1 - \text{sen}(\theta_1) + e(k) & , \text{ se } k = 1..10 \\ \theta_2 - \theta_1 - \text{sen}(\theta_1 + 1) + e(k) & , \text{ se } k = 10..20 \end{cases} \quad (6.28)$$

onde: $y(k)$ é o vetor de medidas.

θ_1 e θ_2 são os parâmetros desconhecidos a serem estimados.

$e(k)$ representa um ruído desconhecido mas limitado por $|e(k)| \leq 1$.

k	y(k)	k	y(k)	k	y(k)	k	y(k)
1	0.41	6	-0.37	11	1.48	16	0.75
2	0.23	7	-0.46	12	1.30	17	0.37
3	0.11	8	-0.59	13	1.11	18	0.56
4	0.15	9	-0.76	14	1.27	19	0.41
5	-0.21	10	-0.76	15	0.93	20	0.22

Tabela 6.5. Vetor de Medidas Para o Modelo Não-Linear

A Tabela 6.5 fornece as medidas simuladas para o exemplo. Neste problema, a rede de Hopfield modificada, combinada com a perspectiva eixos-direcionados, foi utilizada para calcular os intervalos de incerteza paramétrica. O espaço paramétrico inicial para este problema, implementado através da função de ativação rampa-simétrica, é dado por:

$$\Omega = \begin{cases} 0 \leq \theta_1 \leq 12 \\ 1 \leq \theta_2 \leq 11 \end{cases}$$

Na Figura 6.4, a região hachurada representa o conjunto de pertinência paramétrica exata para o problema de identificação robusta definida pelas equação (6.28). Nota-se, para este exemplo, que a região de pertinência paramétrica é formada por duas sub-regiões desconectadas.

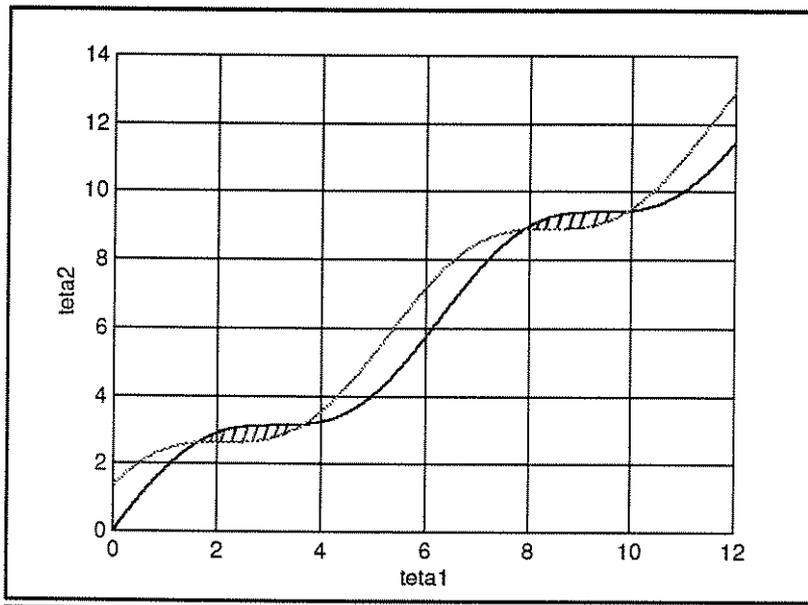


Figura 6.4. Região de Pertinência Paramétrica (Modelo Não-Linear)

Para obter os intervalos de incerteza paramétrica, correspondentes a primeira sub-região, o espaço paramétrico inicial foi reduzido para:

$$\Omega_1 = \begin{cases} 1 \leq \theta_1 \leq 6 \\ 0 \leq \theta_2 \leq 7 \end{cases}$$

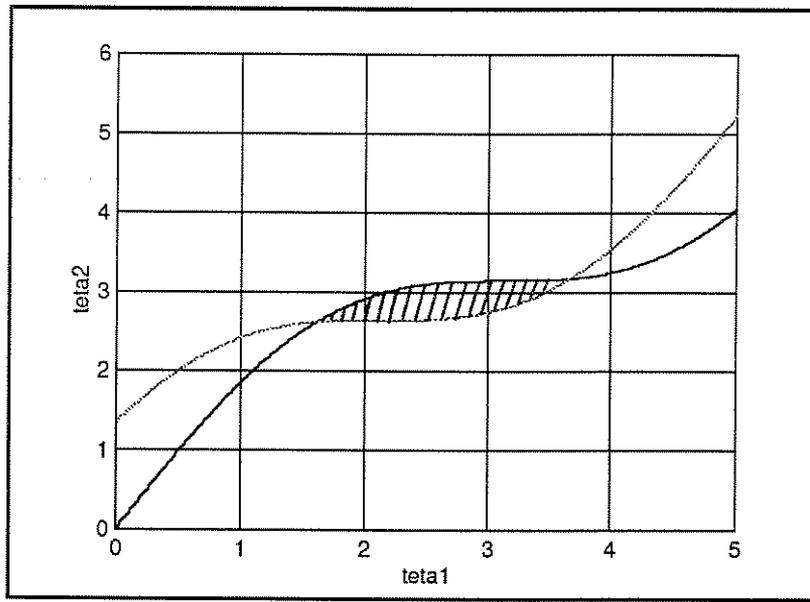


Figura 6.5. Região de Pertinência Paramétrica (1ª Sub-Região)

Assim, os intervalos de incerteza paramétrica calculados pela rede de Hopfield modificada, correspondentes à primeira sub-região, são dados por (Figura 6.5):

$$\begin{aligned} \theta_1^m &= 1.5575 & \theta_1^M &= 3.7257 \\ \theta_2^m &= 2.6120 & \theta_2^M &= 3.1452 \end{aligned}$$

Para os intervalos de incerteza paramétrica, correspondentes à segunda sub-região, o espaço paramétrico foi reduzido para:

$$\Omega_2 = \begin{cases} 6 \leq \theta_1 \leq 12 \\ 8 \leq \theta_2 \leq 11 \end{cases}$$

Os intervalos de incerteza paramétrica, calculados pela rede de Hopfield modificada, para esta sub-região são dados pelos seguintes valores:

$$\begin{aligned} \theta_1^m &= 7.8886 & \theta_1^M &= 9.9551 \\ \theta_2^m &= 8.8983 & \theta_2^M &= 9.4231 \end{aligned}$$

Finalmente, os estimadores centrais, para cada sub-região, calculados a partir dos respectivos intervalos de incerteza paramétrica, são fornecidos na Tabela 6.6.

Estimador Central	1ª Sub-Região	2ª Sub-Região
θ_1^c	2.6416	8.9218
θ_2^c	2.8786	9.1607

Tabela 6.6. Estimadores Centrais (Modelo Não-Linear)

Nos modelos não-lineares, vale ressaltar que quando não há nenhuma informação sobre as sub-regiões que porventura compõem a região de pertinência paramétrica, a rede de Hopfield modificada sempre converge para a sub-região que esteja mais próxima ao ponto onde a rede foi inicializada.

6.5-) CONCLUSÃO

Neste capítulo, a rede de Hopfield modificada foi utilizada na resolução de problemas de identificação paramétrica robusta para modelos lineares e não-lineares. A busca dos pontos de equilíbrio da rede foi conduzida por duas perspectivas diferentes: a perspectiva eixos-direcionados, desenvolvida para calcular o intervalo de incerteza paramétrica para cada parâmetro a ser estimado; e a perspectiva eixos-hipercúbicos, que fornece uma descrição politópica mais exata da região de pertinência paramétrica.

Para modelos lineares, a rede de Hopfield modificada pode simular dois métodos clássicos, ortotopo-externo e politopo-exato, que são aplicados na resolução destes tipos de problemas. Em contraste com o método politopo-exato, que é fortemente dependente da configuração do politopo inicial [Mo90], a rede de Hopfield modificada não requer nenhum procedimento especial para a sua inicialização.

Para modelos não-lineares, a rede proposta é aplicada para calcular os intervalos de incerteza paramétrica de cada estimador. Neste caso, quando a região de pertinência paramétrica é formada por sub-regiões, a rede de Hopfield modificada sempre encontra um intervalo de incerteza que corresponde a uma destas sub-regiões. Assim, pode-se deduzir inicialmente que pelo menos um estimador de θ^* é encontrado pela rede de Hopfield modificada quando a região de pertinência paramétrica é constituída por várias sub-regiões.

BIBLIOGRAFIA

[Arru92] L. V. R. Arruda. *Etude d'algorithmes d'estimation robuste et développement d'un système à base de connaissance pour l'identification*. Thèse de Doctorat, Université de Nice, Sophia Antipolis, France, 1992.

- [Belf88] G. Belforte, B. Bona and V. Cerone. "Parameter estimation with set membership uncertainty: Nonlinear families of models". *Proc. of the 8th IFAC/IFORS Symp. on Identification and System Parameter Estimation*, Beijing, pp. 399-404, 1988.
- [Bill80] S. A. Billings. "Identification of nonlinear systems - A Survey". *IEE PROC.*, vol. 127, Pt. D, no. 6, pp. 272-285, 1980.
- [Clem88] T. Clement and S. Gentil. "Reformulation of parameter identification with unknown-but-bounded errors". *Mathematics and Computers in Simulation*, vol. 30, pp. 257-270, 1988.
- [Foge82] E. Fogel and Y. F. Huang. "On the value of information in system identification: bounded noise case". *Automatica*, vol. 18, no. 2, pp. 229-238, 1982.
- [Kace86] B. Z. Kacewics, M. Milanese, R. Tempo, and A. Vicino. "Optimality of central and projection algorithms for bounded uncertainty". *Syst. Control Letters*, vol. 8, pp. 161-171, 1986.
- [Kees87] K. J. Keesman and G. van Straten. "Modified set theoretic identification of an ill-defined water quality system from poor data". *Proc. of the IAWPRC Symp. Systems Analysis in Water Quality Management*, Pergamon Press, pp. 297-308, 1987.
- [Ljun89] L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [Mila85] M. Milanese and R. Tempo. Optimal algorithms theory for robust estimation and prediction. *IEEE Trans. on Automatic Control*, vol. AC-30, no. 8, 1985.
- [Mo90] S. H. Mo and J. P. Norton. "Fast and robust algorithm to compute exact polytope parameter bounds". *Mathematics and Computers in Simulation*, vol. 32, pp. 481-493, 1990.
- [Silv95] I. N. da Silva. *Estimação Paramétrica Robusta Através de Redes Neurais Artificiais*. Tese de Mestrado, DCA/FEE/UNICAMP, Campinas-SP, Janeiro de 1995.
- [Silv97] I. N. da Silva, L. V. R de Arruda, W. C. do Amaral. "Robust estimation of parametric membership regions using artificial neural networks". *International Journal of Systems Science*, vol. 28, no. 5, pp. 447-455, 1997.

[Vici89] A. Vicino and M. Milanese. "Optimal inner bounds of feasible parameter set in linear estimation with bounded noise". *Proc. of the 28th IEEE Conf. on Decision and Control*, Tampa, USA, pp. 2576-2580, 1989.

[Walt90] E. Walter and H. Piet-Lahanier, "Estimation of parameter bounds from bounded-error data: a survey". *Mathematics and Computers in Simulation*, vol. 32, pp. 449-468, 1990.

CAPÍTULO 7

CONCLUSÕES GERAIS E TRABALHOS FUTUROS

7.1-) RESUMO DA TESE

O presente trabalho concentrou-se no desenvolvimento de uma arquitetura de redes neurais artificiais para solucionar problemas de otimização de sistemas e identificação robusta. Mais especificamente, desenvolveu-se uma rede de Hopfield modificada que foi aplicada no mapeamento dos seguintes problemas: o problema de otimização linear, o problema de otimização quadrática, o problema de otimização combinatorial, o problema de programação dinâmica, o problema de otimização não-linear e problemas de identificação robusta.

Particularmente, para problemas de otimização combinatorial, utilizou-se a rede desenvolvida para resolver o problema do caixeiro viajante, o problema das N-rainhas e o problema do matching-bipartido. Para problemas de programação dinâmica, a rede foi utilizada para resolver o problema do caminho mínimo. Uma função objetivo, correspondente ao termo de otimização, foi desenvolvida com a finalidade de obter o menor caminho entre todos os possíveis. Em relação aos problemas de otimização não-linear, aplica-se a abordagem desenvolvida na solução de problemas genéricos de otimização restrita e irrestrita. O mapeamento destes problemas foi realizado usando a mesma metodologia proposta no Capítulo 3 para tratar problemas de programação linear. Utilizando-se os conceitos básicos e definições da teoria de otimização não-linear e da teoria de estabilidade de Lyapunov, obteve-

se os parâmetros internos da rede. Nos problemas de identificação paramétrica robusta, a rede foi utilizada para estimar os parâmetros desconhecidos de modelos lineares e os intervalos de incerteza paramétrica para os modelos não-lineares.

Todos os problemas abordados nesta tese foram implementados em microcomputador utilizando o software científico MATLAB[†].

7.2-) CONCLUSÕES GERAIS

A abordagem desenvolvida permitiu que diferentes problemas de otimização e identificação robusta fossem solucionados, através da rede de Hopfield modificada, a partir de uma mesma sistemática de mapeamento. Uma das principais vantagens de se utilizar a arquitetura proposta é a de que uma única arquitetura pode ser utilizada para resolver problemas de diferentes naturezas.

Os problemas, tratados nesta tese, são mapeados pela rede de Hopfield desenvolvida através da definição de duas funções de energia denotadas por E^{ot} e E^{conf} . O termo de energia E^{ot} está intrinsecamente ligado à minimização do critério de otimalidade relacionado ao problema a ser mapeado. A minimização do termo de energia E^{conf} , através do subespaço-válido, garante que as soluções (pontos de equilíbrio) encontradas pela rede satisfazem as restrições estruturais impostas para cada problema. Portanto, a técnica do subespaço-válido permite agregar, de forma simples e compacta, todas as restrições envolvidas em um determinado problema.

Os problemas de otimização, mapeados e solucionados através da rede de Hopfield modificada, foram comparados com outros métodos utilizados para resolver cada tipo de problema. As diversas simulações realizadas confirmam que a rede de Hopfield modificada é uma alternativa viável às redes neurais utilizadas na literatura para solucionar os problemas de otimização apresentados.

O controlador lógico nebuloso, associado à rede de Hopfield modificada, permitiu obter-se um método eficiente de conduzir a convergência da rede em direção aos pontos de equilíbrio. Como resultado, o tempo de convergência da rede é reduzido e a precisão dos valores dos pontos de equilíbrio são sensivelmente melhorados.

[†] MATLAB for Windows 4.2b (Math Works Incorporation, 1994)

Vale ressaltar também que uma das vantagens em se utilizar a rede proposta, quando comparada com outras abordagens de redes, é que a rede de Hopfield modificada não necessita da definição de nenhum parâmetro (ponderação ou controle) para a sua inicialização.

Como principal desvantagem da rede proposta deve-se citar a característica não recursiva (off-line) de sua arquitetura, dificultando assim sua implementação em tempo-real.

Enfatiza-se a seguir as principais conclusões, relativas a cada tipo de problema, destacadas previamente em cada capítulo.

Problemas de Otimização Linear e Quadrática

Para os problemas de otimização linear e quadrática, uma análise de complexidade da rede proposta, relativa a outras abordagens, constata que a rede proposta é tratável computacionalmente. Resultados de simulações mostraram que a abordagem proposta apresenta bom desempenho quando comparada com outras arquiteturas de redes neurais. Estes resultados ilustram que a rede desenvolvida pode também ser utilizada como uma alternativa viável em relação às outras abordagens neurais propostas na literatura.

Problemas de Otimização Não-Linear

Nos problemas de otimização não-linear, aplicou-se a abordagem desenvolvida na solução de problemas de otimização restritos e irrestritos. Os problemas de otimização irrestritos foram tratados como um caso particular de problemas restritos. Os resultados de simulações mostraram que a rede proposta fornece bons resultados, tanto para problemas de otimização restritos como para problemas irrestritos. Destes resultados, considera-se que a rede desenvolvida pode também ser utilizada como uma alternativa viável às arquiteturas de redes neurais que utilizam multiplicadores de Lagrange.

Problemas de Otimização Combinatorial e Programação Dinâmica

Para os problemas de otimização combinatorial, utilizou-se a rede desenvolvida para resolver o problema do caixeiro viajante, o problema das N-rainhas e o problema do matching-bipartido. Para os problemas de programação dinâmica, a rede de Hopfield modificada foi aplicada na solução do problema do caminho mínimo. As principais vantagens em utilizar a

rede de Hopfield modificada em relação às outras redes, também utilizadas em otimização combinatorial e programação dinâmica são: (i) não necessidade de especificação de parâmetros de ponderação; (ii) as restrições estruturais do problema são diretamente mapeadas através do subespaço-válido; (iii) o termo de otimização e os termos de restrições da função de energia são tratados em estágios distintos.

Problemas de Identificação Robusta

Para os problemas de identificação robusta, duas perspectivas de busca dos pontos de equilíbrio foram desenvolvidas. A perspectiva eixos-direcionados permite o cálculo dos intervalos de incerteza paramétrica para cada parâmetro a ser estimado, enquanto a perspectiva eixos-hipercúbicos permite uma melhor descrição da região de pertinência paramétrica. Nos modelos lineares, a rede de Hopfield modificada pode simular dois métodos clássicos, ortotopo-externo e politopo-exato, que são também aplicados na resolução destes tipos de problemas. Para os modelos não lineares, a rede proposta é aplicada para calcular os intervalos de incerteza paramétrica para cada estimador. Neste caso, quando a região de pertinência for formada por sub-regiões, a rede de Hopfield modificada sempre encontra um intervalo de incerteza que corresponde a uma destas sub-regiões.

7.3-) LINHAS FUTURAS DE PESQUISAS

Entre os trabalhos futuros, destacam-se a seguir três linhas de pesquisa como continuação deste trabalho.

A primeira consiste em aplicar a rede desenvolvida em novos tipos de problemas de otimização, objetivando agregar um número maior de problemas que podem ser mapeados através da arquitetura propiciada pela rede de Hopfield modificada. Particularmente, pretende-se investigar o uso da abordagem proposta em problemas de programação dinâmica probabilística, e otimização H_2 e H_∞ .

A segunda linha de pesquisa, relacionada com a parte de identificação robusta para modelos não-lineares, consiste no desenvolvimento de métodos que possibilitem a quantificação do número de sub-regiões desconectadas que constituem a região de pertinência paramétrica.

A terceira linha de pesquisa consiste em analisar as modificações e implicações na implementação em hardware da rede de Hopfield modificada para resolver estes tipos de problemas.

APÊNDICE A

Uma das razões principais para a popularidade da rede de Hopfield é a possibilidade de implementá-la em *hardware*. Quando Hopfield e Tank propuseram a rede contínua, eles também demonstraram que existe um hardware analógico equivalente muito simples (Figura A.1). Neste hardware, os neurônios são modelados através de amplificadores operacionais com relação entrada-saída determinada pela função de ativação neuronal.

Cada neurônio possui um resistor R_i^{in} e um capacitor C_i^{in} de entrada que definem uma constante de tempo dos neurônios e estabelece a soma analógica integrativa das correntes de entrada vindas de outros neurônios da rede. Uma conexão entre dois neurônios é definida por uma condutância T_{ij} que conecta uma das duas saídas do amplificador 'j' a entrada do amplificador 'i'. Esta conexão é feita com um resistor de valor $R_{ij} = 1/|T_{ij}|$. Se a sinapse (conexão) for excitatória ($T_{ij} \geq 0$), este resistor será conectado a saída normal (+) do amplificador 'j'. Para uma sinapse inibitória ($T_{ij} < 0$), é conectada a saída inversa (-) do amplificador 'j'. A matriz \mathbf{T} define então a conectividade entre os amplificadores operacionais (neurônios). A corrente de entrada em qualquer neurônio 'i' é a soma das correntes fluindo através do conjunto de resistores que conecta suas entradas às saídas dos outros neurônios.

Como indicado na Figura A.1, este circuito também inclui uma corrente de entrada I_i fornecida externamente para cada neurônio.

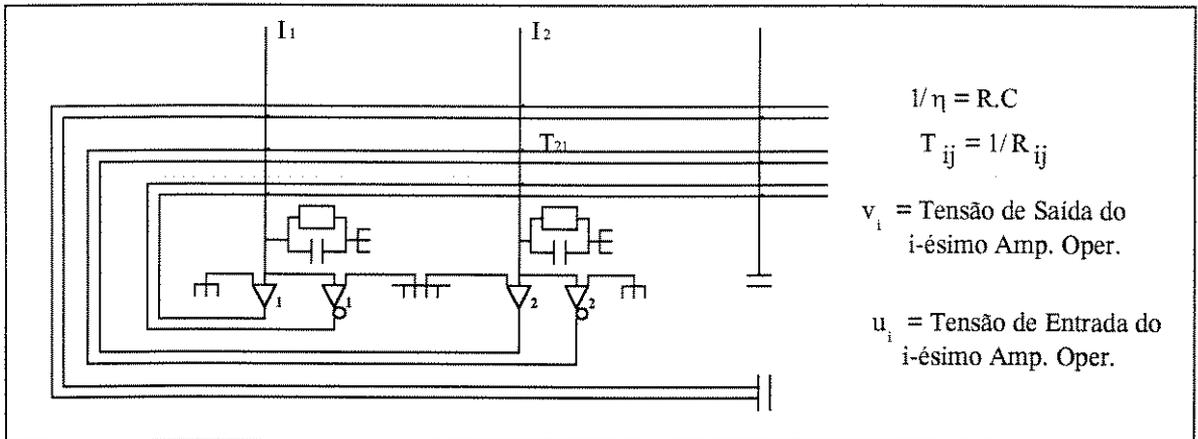


Figura A.1. Hardware Analógico da Rede de Hopfield

As equações que descrevem a evolução deste circuito em relação ao tempo é definida por:

$$C_i \left(\frac{du_i}{dt} \right) = \sum_{j=1}^N T_{ij} \cdot V_j - \frac{u_i}{R_i} + I_i \tag{A.1}$$

$$V_j = g_j(u_j)$$

Define-se o termo R_i como a associação em paralelo de R_i^{in} e os termos R_{ij} , ou seja:

$$\frac{1}{R_i} = \frac{1}{R_i^{in}} + \sum_{j=1}^N \left(\frac{1}{R_{ij}} \right) \tag{A.2}$$

Assumindo que todos os neurônios possuem as mesmas configurações, ou seja, $g_i=g$, $R_i=R$ e $C_i=C$; e dividindo o segundo membro da equação (A.1) por C e redefinindo T_{ij}/C e I_i/C como T_{ij} e I_i respectivamente, a equação (A.1) torna-se:

$$\frac{du_i}{dt} = -\eta \cdot u_i + \sum_{j=1}^N T_{ij} \cdot V_j + I_i \tag{A.3}$$

$\eta = 1 / RC$ e $V_j = g(u_j)$. Vale notar que a equação (A.3) é idêntica a equação (2.1).

A equação (A.3) fornece uma descrição completa da evolução temporal do estado do circuito. A integração desta equação em computador digital permite a simulação de qualquer configuração hipotética da rede de Hopfield.

APÊNDICE B

A análise realizada no capítulo 2 reflete a convergência dinâmica da rede de Hopfield modificada. Com a finalidade de obter um melhor entendimento sobre a função do vetor \mathbf{b} , pode-se associar \mathbf{b} a um problema linear auxiliar. Inicialmente, reescreve-se a função objetivo do problema (2.6) quando \mathbf{v} estiver sobre o subespaço-válido (2.9):

$$E^{ot} = -\frac{1}{2}(\mathbf{T}^{val}\mathbf{v} + \mathbf{s})^T \mathbf{T}^{ot} (\mathbf{T}^{val}\mathbf{v} + \mathbf{s}) - (\mathbf{i}^{ot})^T (\mathbf{T}^{val}\mathbf{v} + \mathbf{s}) \quad (\text{B.1})$$

Simplificando-se a equação (B.1), obtém-se:

$$E^{ot} = -\frac{1}{2}\mathbf{v}^T \mathbf{A} \mathbf{v} - \mathbf{b}^T \mathbf{v} \quad (\text{B.2})$$

Assim, pode-se associar a ação de \mathbf{b} com a minimização de um problema linear auxiliar E^{aux} com função objetivo dada por:

$$E^{aux} = -\mathbf{b}^T \cdot \mathbf{v} \quad (\text{B.3})$$

A dinâmica global da rede, reescrita em termos de \mathbf{A} e \mathbf{b} , são:

$$\dot{\mathbf{v}} = \dot{\mathbf{v}}^{val} = -\nabla E^{ot} = \mathbf{A} \mathbf{v} + \mathbf{b} \quad (\text{B.4})$$

No início da convergência, quando \mathbf{v} está próximo ao centro do subespaço-válido (isto é: $\mathbf{v} \approx \mathbf{s}$), tem-se:

$$\begin{aligned} \mathbf{A} \mathbf{v} &\approx \mathbf{0} \Rightarrow \dot{\mathbf{v}} \approx \mathbf{b} \\ \mathbf{b} &= -\nabla E^{aux} \end{aligned}$$

Logo, conclui-se que o comportamento inicial da rede está fortemente relacionado com a minimização de E^{aux} .

APÊNDICE C

Este apêndice fornece uma breve descrição das redes mencionadas em alguns capítulos desta tese.

C.1-) A Rede de Tank e Hopfield (Programação Linear)

A rede proposta por Tank e Hopfield resolve problemas de programação linear representados da seguinte forma:

$$\text{Minimizar } \mathbf{c}^T \cdot \mathbf{v} \quad (\text{C.1})$$

$$\begin{aligned} \text{sujeito a: } f_1(v_1, v_2, \dots, v_N) &\geq 0 \\ f_2(v_1, v_2, \dots, v_N) &\geq 0 \\ &\dots \\ f_p(v_1, v_2, \dots, v_N) &\geq 0 \end{aligned} \quad (\text{C.2})$$

Nesta rede, cada restrição de igualdade deve ser transformada em duas restrições de desigualdade. Logo, o número total de restrições para esta rede é $(2M^{\text{eq}} + M^{\text{in}})$.

A equação que sintetiza o modelo de Tank e Hopfield é dada por:

$$\dot{v}_k = C_k^{-1} \cdot \left[c_k - \frac{1}{R} v_k - \nabla \mathbf{f}_k^T \cdot \mathbf{g}(\mathbf{v}) \right] \quad (\text{C.3})$$

onde: C_k e R são componentes que definem a configuração dos neurônios (Apêndice A).

$$\nabla \mathbf{f}_k^T = \left[\frac{\partial f_1}{\partial v_k}, \frac{\partial f_2}{\partial v_k}, \dots, \frac{\partial f_{p^+}}{\partial v_k} \right], \quad p^+ = 2M^{\text{eq}} + M^{\text{in}} \quad (\text{C.4})$$

$$\mathbf{g}(\mathbf{v}) = [g_1(\mathbf{v}), g_2(\mathbf{v}), \dots, g_{p^+}(\mathbf{v})]^T, \quad \text{e } g_j(\mathbf{v}) = \begin{cases} -f_j(\mathbf{v}) & \text{se } f_j(\mathbf{v}) < 0 \\ 0 & \text{se } f_j(\mathbf{v}) \geq 0 \end{cases} \quad (\text{C.5})$$

Admitindo-se que C_k tenha valor unitário para todos os neurônios, a complexidade neural (por neurônio) para este modelo é $(2 + (2M^{\text{eq}} + M^{\text{in}}))$ tanto para as multiplicações como

para as adições/subtrações. Assim, a complexidade assintótica dos neurônios vale $\Theta(M^{eq} + M^{in})$.

A rede de Tank e Hopfield é constituída por N-neurônios, então a complexidade do modelo é $(N(2 + (2M^{eq} + M^{in})))$, e a complexidade assintótica do modelo vale $\Theta(N(M^{eq} + M^{in}))$.

C.2-) A Rede de Kennedy e Chua (Programação Linear e Quadrática)

A rede proposta por Kennedy e Chua, implementada através de um circuito elétrico denominado circuito de programação não-linear canônico (Figura C.1), considera o seguinte problema de programação não-linear:

$$\begin{aligned} &\text{Minimizar } \phi(v_1, v_2, \dots, v_N) && \text{(C.6)} \\ &\text{sujeito a: } f_1(v_1, v_2, \dots, v_N) \geq 0 \\ &\quad f_2(v_1, v_2, \dots, v_N) \geq 0 \\ &\quad \dots \\ &\quad f_p(v_1, v_2, \dots, v_N) \geq 0 \end{aligned}$$

As equações do circuito para esta rede são dadas por:

$$C_i \frac{dv_i}{dt} = -\frac{\partial \phi}{\partial v_i} - \sum_{j=1}^p i_j \frac{\partial f_j}{\partial v_i} \quad \text{(C.7)}$$

onde $i_j = s.g_j(\cdot)$, sendo que $s > 0$ é um parâmetro de ponderação e $g_j(\cdot)$ é uma função relacionada ao problema considerado. Para o problema de programação linear, a equação (C.6) passa a ser representada por:

$$\text{Minimizar } \phi(v_1, v_2, \dots, v_N) = \mathbf{c}^T \cdot \mathbf{v} \quad \text{(C.8)}$$

onde $\mathbf{c} \in \mathbb{R}^N$. Neste caso, o parâmetro C_i em (C.7) possui também valor unitário.

Para a rede de Kennedy e Chua, cada restrição de igualdade do problema também deve ser transformada em duas restrições de desigualdade. Logo, o número total de restrições para esta rede é $(2M^{eq} + M^{in})$.

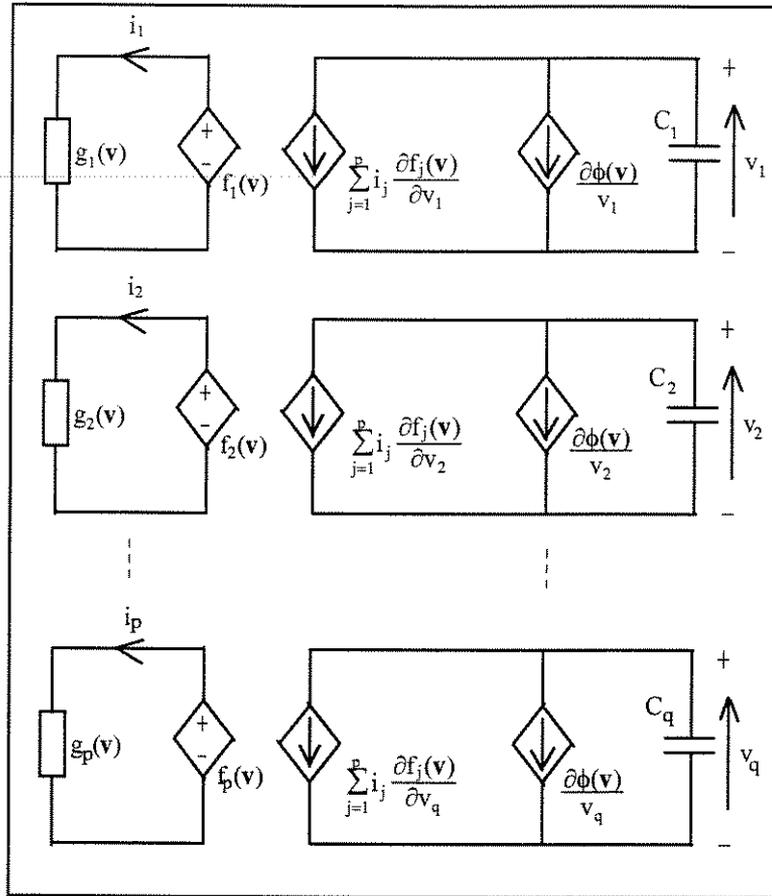


Figura C.1. A Rede de Kennedy e Chua

A equação que sintetiza o modelo de Kennedy e Chua, para problemas de programação linear, é representada por:

$$\dot{v}_k = -c_k - s \cdot \nabla \mathbf{f}_k^T \cdot \mathbf{g}(\mathbf{v})$$

onde: $\nabla \mathbf{f}_k^T$ é definida pela equação (C.4).

$\mathbf{g}(\mathbf{v})$ é definida pela equação (C.5).

A complexidade neural (por neurônio) para este modelo é $(1 + (2M^{eq} + M^{in}))$ tanto para as multiplicações como para as adições/subtrações. Assim, a complexidade assintótica dos neurônios é $\Theta(M^{eq} + M^{in})$.

A rede de Kennedy e Chua é constituída por N-neurônios, então a complexidade do modelo vale $\Theta(N(M^{eq} + M^{in}))$. Convém notar que $\mathbf{g}(\mathbf{v})$, utilizada por todos os neurônios, deve ser calculada apenas uma vez em cada iteração. Entretanto, este cálculo não interfere na complexidade assintótica do modelo, visto que ele é feito utilizando $(2M^{eq} + M^{in})$ comparações.

Das equações acima, nota-se que para a inicialização desta rede, além do passo de iteração, é necessário também escolher o valor do parâmetro 's'.

C.3-) A Rede de Rodríguez-Vázquez et al. (Programação Linear)

A rede proposta por Rodríguez-Vázquez et al. para resolver problemas de programação linear, definidos pelas equações (C.1) e (C.2), é semelhante à rede de Kennedy e Chua. A equação que sintetiza este modelo é dada por:

$$\dot{v}_k = -u(\mathbf{v}) \cdot c_k - s \cdot (1 - u(\mathbf{v})) \cdot \nabla \mathbf{f}_k^T \cdot \mathbf{x}(\mathbf{v}) \quad (\text{C.8})$$

$$\text{onde: } x_j(\mathbf{v}) = \begin{cases} 1, & \text{se } f_j(\mathbf{v}) \geq 0 \\ 0, & \text{se } f_j(\mathbf{v}) < 0 \end{cases}$$

$$u(\mathbf{v}) = \begin{cases} 1, & \text{se } f_j(\mathbf{v}) \geq 0 \text{ para todo } j = 1..(2M^{eq} + M^{in}) \\ 0, & \text{caso contrario} \end{cases}$$

A complexidade neural para este modelo é $(3 + (2M^{eq} + M^{in}))$ para as multiplicações e $(2 + (2M^{eq} + M^{in}))$ para as adições/subtrações. Assim, a complexidade assintótica dos neurônios é $\Theta(M^{eq} + M^{in})$ tanto para as multiplicações como para as adições/subtrações.

A rede de Rodríguez-Vázquez é também constituída por N-neurônios. A complexidade assintótica do modelo vale portanto $\Theta(N(M^{eq} + M^{in}))$. Convém notar que $u(\mathbf{v})$ e $\mathbf{x}(\mathbf{v})$, utilizados por todos os neurônios, devem ser calculados apenas uma vez em cada iteração. Entretanto, estes cálculos não interferem na complexidade assintótica do modelo, visto que eles são feitos utilizando apenas comparações.

Como na rede de Kennedy e Chua, esta rede requer a escolha de Δt e 's' para sua inicialização.

C.4-) A Rede de Zak et al. (Programação Linear)

A rede proposta por Zak et al. resolve problemas de programação linear representados da seguinte forma:

Minimizar $\mathbf{c}^T \cdot \mathbf{v}$

sujeito a: $\mathbf{A} \cdot \mathbf{v} = \mathbf{b}$

A incorporação de restrições de desigualdade no problema de programação linear deve ser feita através da inserção de variáveis auxiliares como em (3.10). Neste caso, o problema acima passa a ser descrito por:

Minimizar $\tilde{\mathbf{c}}^T \cdot \mathbf{v}$

sujeito a: $\tilde{\mathbf{A}} \cdot \tilde{\mathbf{v}} = \tilde{\mathbf{b}}$

onde: $\tilde{\mathbf{A}} \in \mathfrak{R}^{(M^{eq}+M^{in}) \times (N+M^{in})}$; $\tilde{\mathbf{c}}, \tilde{\mathbf{v}} \in \mathfrak{R}^{(N+M^{in})}$ e $\tilde{\mathbf{b}} \in \mathfrak{R}^{(M^{eq}+M^{in})}$.

A equação que sintetiza este modelo é dada por:

$$\tilde{v}_k = -\tilde{c}_k - \rho \cdot \tilde{\mathbf{A}}_k \cdot \mathbf{y}(\tilde{\mathbf{v}}) - \gamma \cdot x_k(\tilde{\mathbf{v}}) \quad (\text{C.9})$$

onde: $\tilde{\mathbf{A}}_k$ é a k-ésima linha de $\tilde{\mathbf{A}}$.

$\mathbf{y}(\tilde{\mathbf{v}}) = [\text{sgn}(\tilde{\mathbf{A}}_1 \cdot \tilde{\mathbf{v}} - \tilde{\mathbf{b}}_1), \text{sgn}(\tilde{\mathbf{A}}_2 \cdot \tilde{\mathbf{v}} - \tilde{\mathbf{b}}_2), \dots, \text{sgn}(\tilde{\mathbf{A}}_{N^+} \cdot \tilde{\mathbf{v}} - \tilde{\mathbf{b}}_{N^+})]$, onde $N^+ = M^{eq} + M^{in}$

$$x_k(\tilde{\mathbf{v}}) = \begin{cases} -1 & , \text{ se } \tilde{v}_k < 0 \\ 0 & , \text{ se } \tilde{v}_k > 0 \end{cases}$$

ρ e γ são parâmetros de ponderação, que devem ser escolhidos a priori durante a inicialização da rede.

O termo \mathbf{y} , utilizado por todos os neurônios, necessita ser calculado apenas uma vez em cada iteração. Este custo é levado em conta no cálculo da complexidade assintótica do modelo.

A complexidade neural (por neurônio) para este modelo é $(2 + (M^{eq} + M^{in}))$ tanto para as multiplicações como para as adições/subtrações. Logo, a complexidade assintótica dos neurônios vale $\Theta(M^{eq} + M^{in})$.

Dado que esta rede é constituída por $(N+M^{in})$ neurônios, então a complexidade do modelo é $(N+M^{in}) \cdot (2+(M^{eq}+M^{in}))$. Deve-se adicionar à complexidade do modelo, o custo do cálculo de \mathbf{y} que é $(N+M^{in}) \cdot (N+M^{eq}+M^{in})$. Logo, a complexidade do modelo torna-se $(N+M^{in}) \cdot (2+N+2M^{eq}+2M^{in})$, e a complexidade assintótica do modelo é dada por $\Theta((N+M^{in})(N+M^{eq} + M^{in}))$.

C.5-) A Rede de Hopfield e Tank (Problema do Caixeiro Viajante)

A rede proposta por Hopfield e Tank, para resolver o problema do caixeiro viajante, tem uma função de energia (que deve ser minimizada) dada por:

$$E(\mathbf{v}) = \frac{A}{2} \sum_x \sum_i \sum_{j \neq i} v_{xi} v_{xj} + \frac{B}{2} \sum_i \sum_x \sum_{y \neq x} v_{xi} v_{yi} + \frac{C}{2} \left(\sum_x \sum_i v_{xi} - N \right)^2 + \frac{D}{2} \sum_x \sum_{y \neq x} \sum_i d_{xy} v_{xi} (v_{y(i+1)} + v_{y(i-1)})$$

onde: \mathbf{v} é a saída da rede.

N é o número de neurônios.

A, B, C e D são parâmetros de ponderação.

C.6-) A Rede de Mandziuk e Macukow (Problema das N-Rainhas)

A rede proposta por Mandziuk e Macukow, para resolver o problema das N-Rainhas, é também baseada no modelo de Hopfield. A função de energia associada à rede, e que deve ser minimizada, é dada por:

$$2E = A \sum_{i=1}^N \sum_{j=1}^N \left[\sum_{\substack{k=1 \\ k \neq j}}^N v_{ik} \right] v_{ij} + \left(\sum_{\substack{k=1 \\ k \neq i}}^N v_{kj} \right) v_{ij} + B \sum_{i=2}^N \sum_{j=1}^{i-1} \left[\left(\sum_{\substack{k=i-j+1 \\ k \neq i}}^N v_{k, k-i+j} \right) \cdot v_{ij} \right] + B \sum_{i=1}^N \sum_{j=1}^N \left[\left(\sum_{\substack{k=1 \\ k \neq i}}^{N+i-j} v_{k, k-i+j} \right) \cdot v_{ij} \right] + B \sum_{i=1}^N \sum_{j=N-i+1}^N \left[\left(\sum_{\substack{k=i+j-N \\ k \neq i}}^N v_{k, i+j-k} \right) \cdot v_{ij} \right] + B \sum_{i=1}^{N-1} \sum_{j=1}^{N-i} \left[\left(\sum_{\substack{k=1 \\ k \neq i}}^{i+j-1} v_{k, i+j-k} \right) \cdot v_{ij} \right] + C \left(\sum_{i=1}^N \sum_{j=1}^N v_{ij} - D \right)^2$$

onde: \mathbf{v} é o vetor de saída da rede.

A, B, C e D são parâmetros de ponderação.

C.7-) A Rede de Chiu et al. (Programação Dinâmica)

A rede proposta por Chiu et al., para resolver problemas de programação dinâmica, é também baseada no modelo de Hopfield. A função de energia desta rede, a ser minimizada, é dada por:

$$E(\mathbf{v}) = \frac{a}{2} E_1(\mathbf{v}) + \frac{b}{4} E_2(\mathbf{v}) + S_1(\mathbf{v}) + S_2(\mathbf{v})$$

onde:

$$E_1(\mathbf{v}) = \sum_x \sum_i \sum_{j \neq i} c_{xi} v_{xi} c_{xj} v_{xj} + \left(\sum_x \sum_i c_{xi} v_{xi} - n \right)^2$$

$$E_2(\mathbf{v}) = \sum_x \sum_i \sum_j d_{xi,(x+1)j} c_{xi} v_{xi} c_{(x+1)j} v_{(x+1)j} + d_{(x-1)j,xi} c_{xi} v_{xi} c_{(x-1)j} v_{(x-1)j}$$

$$\begin{aligned} S_1(\mathbf{v}) &= \frac{a}{2} \sum_x \sum_i \sum_{j \neq i} (1 - c_{xi}) v_{xi} (1 - c_{xj}) v_{xj} + \frac{a}{2} \left(\sum_x \sum_i (1 - c_{xi}) v_{xi} \right)^2 \\ &\quad + \frac{b}{4} \sum_x \sum_i \sum_j d_{xi,(x+1)j} (c_{xi} v_{xi} (1 - c_{(x+1)j}) v_{(x+1)j} + (1 - c_{xi}) v_{xi} (1 - c_{(x+1)j}) v_{(x+1)j}) \\ &\quad + d_{(x-1)j,xi} (c_{xi} v_{xi} (1 - c_{(x-1)j}) v_{(x-1)j} + (1 - c_{xi}) v_{xi} (1 - c_{(x-1)j}) v_{(x-1)j}) \end{aligned}$$

$$\begin{aligned} S_2(\mathbf{v}) &= a \left(\sum_x \sum_i c_{xi} v_{xi} - n \right) \left(\sum_x \sum_i (1 - c_{xi}) v_{xi} \right) \\ &\quad + \frac{a}{2} \sum_x \sum_i \sum_{j \neq i} c_{xi} v_{xi} (1 - c_{xj}) v_{xj} + \frac{a}{2} \sum_x \sum_i \sum_{j \neq i} c_{xj} v_{xj} (1 - c_{xi}) v_{xi} \\ &\quad + \frac{b}{4} \sum_x \sum_i \sum_j d_{xi,(x+1)j} c_{(x+1)j} v_{(x+1)j} (1 - c_{xi}) v_{xi} + \\ &\quad + \frac{b}{4} \sum_x \sum_i \sum_j d_{(x-1)j,xi} c_{(x-1)j} v_{(x-1)j} (1 - c_{xi}) v_{xi} \end{aligned}$$

onde: \mathbf{v} é a saída da rede.

\mathbf{c} é o estado corrente da rede.

‘ n ’ é o número de neurônios da rede.

‘ a ’ e ‘ b ’ são parâmetros de ponderação.

APÊNDICE D

Este apêndice fornece todas as soluções obtidas pela rede de Hopfield modificada para o problema das N-rainhas, com N=5 e N=8. A Tabela D.1 mostra as 8 soluções encontradas para o problema com N=5, e a Tabela D.2 fornece as 92 soluções obtidas pela rede para o problema das N-rainhas com N=8.

n	p₁	p₂	p₃	p₄	p₅
1	1	3	5	2	4
2	1	4	2	5	3
3	2	4	1	3	5
4	3	1	4	2	5
5	3	5	2	4	1
6	4	2	5	3	1
7	5	2	4	1	3
8	5	3	1	4	2

Tabela D1. Soluções Para o Problema das N-Rainhas (N=5)

APÊNDICE E

Este apêndice apresenta uma análise da complexidade da rede de Hopfield modificada em relação aos problemas de otimização combinatorial e programação dinâmica descritos no Capítulo 5. Entretanto, a rede de Hopfield modificada não será comparada com as outras redes, também citadas no Capítulo 5, devido a falta de detalhes de implementação das mesmas.

E.1-) Complexidade do Problema do Caixeiro Viajante

Para o problema do caixeiro viajante com N -cidades, a equação que sintetiza a operação de cada neurônio é dada por:

$$v_k = \Delta t. (\mathbf{T}_k^{\text{ot}} \cdot \mathbf{v} + \mathbf{i}^{\text{ot}}) + (\mathbf{T}_k^{\text{val}} \cdot \mathbf{v} + s_k)$$

onde: \mathbf{T}_k^{ot} corresponde à k -ésima linha de \mathbf{T}^{ot} $\{\mathbf{T}^{\text{ot}} \in \mathfrak{R}^{N^2 \times N^2}\}$.

$\mathbf{T}_k^{\text{val}}$ corresponde à k -ésima linha de \mathbf{T}^{val} $\{\mathbf{T}^{\text{val}} \in \mathfrak{R}^{N^2 \times N^2}\}$.

A complexidade neural (por neurônio) é portanto $(1+2N^2)$ para as multiplicações e $(3+2N^2)$ para as adições/subtrações. Assim, a complexidade assintótica para as multiplicações e adições/subtrações dos neurônios vale $\Theta(N^2)$.

Como o número total de neurônios utilizado para este problema é N^2 , a complexidade do modelo é $N^2(1+2N^2)$ para as multiplicações e $N^2(3+2N^2)$ para as adições/subtrações. Utilizando a definição (3.20), a complexidade assintótica do modelo vale $\Theta(N^4)$ tanto para as multiplicações como para as adições/subtrações.

E.2-) Complexidade do Problema das N-Rainhas

No problema das N -Rainhas, a equação que sintetiza a operação de cada neurônio é dada por:

n	p ₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈	n	p ₁	p ₂	p ₃	p ₄	p ₅	p ₆	p ₇	p ₈
1	1	5	8	6	3	7	2	4	47	5	1	4	6	8	2	7	3
2	1	6	8	3	7	4	2	5	48	5	1	8	4	2	7	3	6
3	1	7	4	6	8	2	5	3	49	5	1	8	6	3	7	2	4
4	1	7	5	8	2	4	6	3	50	5	2	4	6	8	3	1	7
5	2	4	6	8	3	1	7	5	51	5	2	4	7	3	8	6	1
6	2	5	7	1	3	8	6	4	52	5	2	6	1	7	4	8	3
7	2	5	7	4	1	8	6	3	53	5	2	8	1	4	7	3	6
8	2	6	1	7	4	8	3	5	54	5	3	1	6	8	2	4	7
9	2	6	8	3	1	4	7	5	55	5	3	1	7	2	8	6	4
10	2	7	3	6	8	5	1	4	56	5	3	8	4	7	1	6	2
11	2	7	5	8	1	4	6	3	57	5	7	1	3	8	6	4	2
12	2	8	6	1	3	5	7	4	58	5	7	1	4	2	8	6	3
13	3	1	7	5	8	2	4	6	59	5	7	2	4	8	1	3	6
14	3	5	2	8	1	7	4	6	60	5	7	2	6	3	1	4	8
15	3	5	2	8	6	4	7	1	61	5	7	2	6	3	1	8	4
16	3	5	7	1	4	2	8	6	62	5	7	4	1	3	8	6	2
17	3	5	8	4	1	7	2	6	63	5	8	4	1	3	6	2	7
18	3	6	2	5	8	1	7	4	64	5	8	4	1	7	2	6	3
19	3	6	2	7	1	4	8	5	65	6	1	5	2	8	3	7	4
20	3	6	2	7	5	1	8	4	66	6	2	7	1	3	5	8	4
21	3	6	4	1	8	5	7	2	67	6	2	7	1	4	8	5	3
22	3	6	4	2	8	5	7	1	68	6	3	1	7	5	8	2	4
23	3	6	8	1	4	7	5	2	69	6	3	1	8	4	2	7	5
24	3	6	8	1	5	7	2	4	70	6	3	1	8	5	2	4	7
25	3	6	8	2	4	1	7	5	71	6	3	5	7	1	4	2	8
26	3	7	2	8	5	1	4	6	72	6	3	5	8	1	4	2	7
27	3	7	2	8	6	4	1	5	73	6	3	7	2	4	8	1	5
28	3	8	4	7	1	6	2	5	74	6	3	7	2	8	5	1	4
29	4	1	5	8	2	7	3	6	75	6	3	7	4	1	8	2	5
30	4	1	5	8	6	3	7	2	76	6	4	1	5	8	2	7	3
31	4	2	5	8	6	1	3	7	77	6	4	2	8	5	7	1	3
32	4	2	7	3	6	8	1	5	78	6	4	7	1	3	5	2	8
33	4	2	7	3	6	8	5	1	79	6	4	7	1	8	2	5	3
34	4	2	7	5	1	8	6	3	80	6	8	2	4	1	7	5	3
35	4	2	8	5	7	1	3	6	81	7	1	3	8	6	4	2	5
36	4	2	8	6	1	3	5	7	82	7	2	4	1	8	5	3	6
37	4	6	1	5	2	8	3	7	83	7	2	6	3	1	4	8	5
38	4	6	8	2	7	1	3	5	84	7	3	1	6	8	5	2	4
39	4	6	8	3	1	7	5	2	85	7	3	8	2	5	1	6	4
40	4	7	1	8	5	2	6	3	86	7	4	2	5	8	1	3	6
41	4	7	3	8	2	5	1	6	87	7	4	2	8	6	1	3	5
42	4	7	5	2	6	1	3	8	88	7	5	3	1	6	8	2	4
43	4	7	5	3	1	6	8	2	89	8	2	4	1	7	5	3	6
44	4	8	1	3	6	2	7	5	90	8	2	5	3	1	7	4	6
45	4	8	1	5	7	2	6	3	91	8	3	1	6	2	5	7	4
46	4	8	5	3	1	7	2	6	92	8	4	1	3	6	2	7	5

Tabela D2. Soluções Para o Problema das N-Rainhas (N=8)

$$v_k = \Delta t. (\mathbf{T}_k^{\text{ot}} \cdot \mathbf{v}) + (\mathbf{T}_k^{\text{val}} \cdot \mathbf{v} + s_k)$$

onde: \mathbf{T}_k^{ot} corresponde à k-ésima linha de \mathbf{T}^{ot} $\{\mathbf{T}^{\text{ot}} \in \mathfrak{R}^{N^2 \times N^2}\}$.

$\mathbf{T}_k^{\text{val}}$ corresponde à k-ésima linha de \mathbf{T}^{val} $\{\mathbf{T}^{\text{val}} \in \mathfrak{R}^{N^2 \times N^2}\}$.

A complexidade neural é $(1+2N^2)$ para as multiplicações e $(2+2N^2)$ para as adições/subtrações. Logo, a complexidade assintótica para as multiplicações e adições/subtrações dos neurônios é $\Theta(N^2)$. Como o número total de neurônios utilizado para este problema é N^2 , a complexidade assintótica do modelo é $\Theta(N^4)$ tanto para as multiplicações como para as adições/subtrações.

E.3-) Complexidade do Problema do Matching-Bipartido

No problema do matching-bipartido com N-elementos, a equação que sintetiza a operação de cada neurônio é dada por:

$$v_k = \Delta t. i_k^{\text{ot}} + (\mathbf{T}_k^{\text{val}} \cdot \mathbf{v} + s_k)$$

onde: $\mathbf{T}_k^{\text{val}}$ corresponde à k-ésima linha de \mathbf{T}^{val} $\{\mathbf{T}^{\text{val}} \in \mathfrak{R}^{N^2 \times N^2}\}$.

i_k^{ot} corresponde ao k-ésimo elemento de \mathbf{i}^{ot} $\{\mathbf{i}^{\text{ot}} \in \mathfrak{R}^{N^2}\}$.

A complexidade neural é $(1+N^2)$ para as multiplicações e $(2+N^2)$ para as adições/subtrações. Por conseguinte, a complexidade assintótica para as multiplicações e adições/subtrações dos neurônios vale $\Theta(N^2)$. Como o número total de neurônios utilizado é N^2 , a complexidade assintótica do modelo é $\Theta(N^4)$ tanto para as multiplicações como para as adições/subtrações.

E.4-) Complexidade do Problema de Programação Dinâmica

No problema de programação dinâmica (caminho mínimo), com n-estágios e m-estados por estágio, a equação que sintetiza a operação de cada neurônio é dada por:

$$v_k = \Delta t. (\mathbf{T}_k^{\text{ot}} \cdot \mathbf{v} + \mathbf{i}^{\text{ot}}) + (\mathbf{T}_k^{\text{val}} \cdot \mathbf{v} + s_k)$$

onde: \mathbf{T}_k^{ot} corresponde à k-ésima linha de \mathbf{T}^{ot} $\{\mathbf{T}^{\text{ot}} \in \mathfrak{R}^{nm \times nm}\}$.

$\mathbf{T}_k^{\text{val}}$ corresponde à k-ésima linha de \mathbf{T}^{val} $\{\mathbf{T}^{\text{val}} \in \mathfrak{R}^{nm \times nm}\}$.

A complexidade neural é $(1+2n.m)$ para as multiplicações e $(3+2n.m)$ para as adições/subtrações. Assim, a complexidade assintótica para as multiplicações e adições/subtrações dos neurônios é $\Theta(n.m)$.

Como o número total de neurônios utilizado para este problema é $n.m$, a complexidade do modelo é $(n.m).(1+2n.m)$ para as multiplicações e $(n.m).(3+2n.m)$ para as adições/subtrações. Logo, a complexidade assintótica do modelo vale $\Theta(n^2.m^2)$ tanto para as multiplicações como para as adições/subtrações.