

UNIVERSIDADE DE SÃO PAULO
Escola de Engenharia de São Carlos
Programa de Pós-Graduação em Engenharia Elétrica

Fabiana Cristina Bertoni

**Uma Arquitetura Neuro-Genética Para
Otimização Não-Linear Restrita**

São Carlos
Outubro 2007

Fabiana Cristina Bertoni

**Uma Arquitetura Neuro-Genética Para
Otimização Não-Linear Restrita**

*Tese apresentada à Escola de Engenharia de
São Carlos da Universidade de São Paulo,
sendo parte dos requisitos para obtenção do
título de Doutor em Engenharia Elétrica.*

Orientador: Prof. Dr. Ivan Nunes da Silva

**São Carlos
Outubro 2007**

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

Romanos 11,36

*O homem como um lápis...
É possível fazer grandes coisas, mas existe uma mão (Deus) que guia.
Às vezes, é preciso usar o apontador... Sofrimento que leva à perfeição.
Sempre existe a possibilidade de usar a borracha e corrigir os erros.
O que importa é o que está dentro da madeira.
O lápis sempre deixa uma marca.*

Autor desconhecido

Agradecimentos

Agradeço a DEUS, no nome de JESUS, por seu infinito amor. Amor que o levou a dar sua própria vida em meu favor. Sou imensamente grata por tudo o que sou e por tudo o que me deste. Glória eterna ao Seu nome!

Aos meus pais e avó materna que sempre foram exemplo para mim de garra, dedicação e amor. A vocês, por tudo que me ensinaram na vida.

Um agradecimento especial ao professor Ivan Nunes da Silva, que é muito mais do que um orientador, me instruindo com sabedoria, companheirismo e amizade. Reconheço e admiro sua atitude em ajudar seus “desorientados”, dedicando seu tempo e paciência, ajudando-me a crescer na profissão e na vida. Obrigada pelos incentivos, por aceitar ser meu orientador, por seus significativos ensinamentos. Obrigada por tudo.

Aos meus amigos do coração, Luciana Abdo e Evandro Silva, pelo carinho e atenção dedicados a esta “miguinha” maluca.

A todos os amigos do Laboratório de Automação Inteligente de Processos e Sistemas: Spatti e Alessandro (os melhores churrasqueiros que já conheci), Suetake, Sergião, Patrícia, Cabeça ou Louro José (vulgo Rodrigo) e Cristiano, que me apoiaram sempre que necessitei da ajuda de cada um deles.

Por fim, agradeço aos colegas professores do curso de Engenharia da Computação da Universidade Estadual de Feira de Santana, pela amizade, compreensão e apoio ofertados.

Agradecimento Especial...

Ao meu marido, Matheus Giovanni Pires...

Agradeço a Deus cada momento que passo com você, agradeço por ter conhecido a melhor pessoa do mundo, por amar e sonhar cada momento que passamos juntos, os quais pareciam ser eternos. Agradeço a Deus por ser a pessoa mais feliz do mundo, pois tenho você ao meu lado...

Mas, depois de agradecer a nosso maravilhoso Deus, agradeço a você, meu amor... por tudo...

Pelos momentos em que chorei, e que você veio carinhosamente, me beijou e me fez sorrir.

Pelos momentos em que perdi a paciência, e você sempre lá, com palavras amenas e doces para me acalmar.

Por estar comigo em todos os momentos, e fazer com que eu não desistisse.

Pelos momentos de alegria, que fez questão de dividir comigo.

E pelos momentos que com muita esperança, pensou junto comigo o nosso futuro.

Obrigada, meu amor.

Obrigada, por existir em minha vida.

Obrigada, por me fazer feliz e a pessoa mais amada na tua vida.

Te amo!!!

Resumo

BERTONI, F. C. (2007). Uma Arquitetura Neuro-Genética Para Otimização Não-Linear Restrita. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.

Os sistemas baseados em redes neurais artificiais e algoritmos genéticos oferecem um método alternativo para solucionar problemas relacionados à otimização de sistemas. Os algoritmos genéticos devem a sua popularidade à possibilidade de percorrer espaços de busca não-lineares e extensos. As redes neurais artificiais possuem altas taxas de processamento por utilizarem um número elevado de elementos processadores simples com alta conectividade entre si. Redes neurais com conexões realimentadas fornecem um modelo computacional capaz de resolver vários tipos de problemas de otimização, os quais consistem, geralmente, da otimização de uma função objetivo que pode estar sujeita ou não a um conjunto de restrições. Esta tese apresenta uma abordagem inovadora para resolver problemas de otimização não-linear restrita utilizando uma arquitetura neuro-genética. Mais especificamente, uma rede neural de Hopfield modificada é associada a um algoritmo genético visando garantir a convergência da rede em direção aos pontos de equilíbrio factíveis que representam as soluções para o problema de otimização não-linear restrita.

Palavras-Chave: Redes Neurais, Algoritmos Genéticos, Otimização Não-Linear Restrita.

Abstract

BERTONI, F. C. (2007). Neuro-genetic Architecture for Constrained Nonlinear Optimization. Thesis (Doctorate Degree) – Escola de Engenharia de São Carlos, Universidade de São Paulo, 2007.

Systems based on artificial neural networks and genetic algorithms are an alternative method for solving systems optimization problems. The genetic algorithms must its popularity to make possible cover nonlinear and extensive search spaces. Artificial neural networks have high processing rates due to the use of a massive number of simple processing elements and the high degree of connectivity between these elements. Neural networks with feedback connections provide a computing model capable of solving a large class of optimization problems, which refer to optimization of an objective function that can be subject to constraints. This thesis presents a novel approach for solving constrained nonlinear optimization problems using a neuro-genetic approach. More specifically, a modified Hopfield neural network is associated with a genetic algorithm in order to guarantee the convergence of the network to the equilibrium points, which represent feasible solutions for the constraint nonlinear optimization problem.

Keywords: Neural Networks, Genetic Algorithms, Constrained Nonlinear Optimization.

Lista de Figuras

FIGURA 3.1. A REDE NEURAL DE HOPFIELD CONVENCIONAL.....	23
FIGURA 3.2. A REDE DE HOPFIELD GENÉTICA.....	29
FIGURA 4.1: CRUZAMENTO BLX- α	51
FIGURA 4.2. MUTAÇÃO UNIFORME.....	51
FIGURA 5.1: EVOLUÇÃO DO VETOR DE SAÍDA DA RHG.....	55
FIGURA 5.2: COMPORTAMENTO DA FUNÇÃO OBJETIVO.....	55
FIGURA 5.3: VETOR V EM DUAS EXECUÇÕES DIFERENTES.....	56
FIGURA 5.4: VALOR DA FUNÇÃO OBJETIVOS PARA DUAS EXECUÇÕES DIFERENTES.....	57
FIGURA 5.5: EVOLUÇÃO DO VETOR DE SAÍDA DA RHG.....	58
FIGURA 5.6: COMPORTAMENTO DA FUNÇÃO OBJETIVO.....	60
FIGURA 5.7:EVOLUÇÃO DO VETOR DE SAÍDA DA RHG.....	61
FIGURA 5.8: COMPORTAMENTO DA FUNÇÃO OBJETIVO.....	62
FIGURA 5.9: EVOLUÇÃO DO VETOR DE SAÍDA DA RHG.....	63
FIGURA 5.10: COMPORTAMENTO DA FUNÇÃO OBJETIVO.....	64
FIGURA 5.11: EVOLUÇÃO DO VETOR DE SAÍDA DA RHG.....	65
FIGURA 5.12: COMPORTAMENTO DA FUNÇÃO OBJETIVO.....	66
FIGURA 5.13: EVOLUÇÃO DO VETOR DE SAÍDA DA RHG.....	67
FIGURA 5.14: COMPORTAMENTO DA FUNÇÃO OBJETIVO.....	68
FIGURA 5.15: EVOLUÇÃO DO VETOR DE SAÍDA DA RHG.....	69
FIGURA 5.16: COMPORTAMENTO DA FUNÇÃO OBJETIVO.....	70
FIGURA 5.17: EVOLUÇÃO DO VETOR DE SAÍDA DA RHG.....	71
FIGURA 5.18: COMPORTAMENTO DA FUNÇÃO OBJETIVO.....	72
FIGURA 5.19: EVOLUÇÃO DO VETOR DE SAÍDA DA RHG.....	73
FIGURA 5.20: COMPORTAMENTO DA FUNÇÃO OBJETIVO.....	73
FIGURA 5.21: EVOLUÇÃO DO VETOR DE SAÍDA DA RHG.....	75
FIGURA 5.22: COMPORTAMENTO DA FUNÇÃO OBJETIVO.....	75
FIGURA 5.23: REGIÃO DE PERTINÊNCIA PARAMÉTRICA (MODELO NÃO-LINEAR).....	80
FIGURA A. 1: HARDWARE ANALÓGICO DA REDE DE HOPFIELD.....	96
FIGURA A. 2: EVOLUÇÃO DE V - ALGORITMO GENÉTICO (CODIFICAÇÃO BINÁRIA).....	98
FIGURA A. 3: EVOLUÇÃO DE F(V) - ALGORITMO GENÉTICO (CODIFICAÇÃO BINÁRIA).....	98

Lista de Tabelas

TABELA 2.1: ABORDAGENS NEURAIIS APLICADAS EM PROBLEMAS DE OTIMIZAÇÃO RESTRITA.	15
TABELA 2.2: ABORDAGENS EVOLUTIVAS APLICADAS EM PROBLEMAS DE OTIMIZAÇÃO RESTRITA.	18
TABELA 2.3: ABORDAGENS HÍBRIDAS APLICADAS EM PROBLEMAS DE OTIMIZAÇÃO RESTRITA.	19
TABELA 5.1: RESULTADOS UTILIZANDO A RHG.	59
TABELA 5.2: RESULTADOS UTILIZANDO A RHG.	62
TABELA 5.3: VETOR DE MEDIDAS PARA O MODELO NÃO-LINEAR.	79
TABELA 5.4: RESULTADOS DO PROBLEMA DE ESTIMAÇÃO ROBUSTA (1ª SUB-REGIÃO).	80
TABELA 5.5: INTERVALOS DE INCERTEZA PARAMÉTRICA OBTIDOS PELA RHG.	81
TABELA 5.6: RESULTADOS DO PROBLEMA DE ESTIMAÇÃO ROBUSTA (2ª SUB-REGIÃO).	81
TABELA 5.7: INTERVALOS DE INCERTEZA PARAMÉTRICA OBTIDOS PELA RHG.	81
TABELA 5.8: ESTIMADORES CENTRAIS (MODELO NÃO-LINEAR).	82

Lista de Algoritmos

ALGORITMO 4.1: ESTRUTURA DE UM ALGORITMO GENÉTICO	44
ALGORITMO 4.2: MÉTODO DE SELEÇÃO POR TORNEIO	49

Lista de Abreviaturas e Siglas

RNA	Redes Neurais Artificiais
RHG	Rede de Hopfield Genética
AG	Algoritmo Genético
SQP	<i>Sequential Quadratic Programming</i>
ALM-NN	<i>Augmented Lagrange Multiplier – Neural Network</i>

Sumário

RESUMO	VII
ABSTRACT	IX
LISTA DE FIGURAS	XI
LISTA DE TABELAS	XIII
LISTA DE ALGORITMOS	XV
LISTA DE ABREVIATURAS E SIGLAS	XVII
CAPÍTULO 1 INTRODUÇÃO	1
1.1 OBJETIVO E MOTIVAÇÃO	1
1.2 JUSTIFICATIVA E RELEVÂNCIA.....	3
1.3 CONTRIBUIÇÕES DA TESE	4
1.4 ORGANIZAÇÃO DO TRABALHO.....	6
CAPÍTULO 2 ASPECTOS DE SISTEMAS INTELIGENTES APLICADOS EM OTIMIZAÇÃO NÃO-LINEAR	9
2.1 ABORDAGENS DE REDES NEURAIS APLICADAS EM OTIMIZAÇÃO RESTRITA	10
2.2 ABORDAGENS DE COMPUTAÇÃO EVOLUTIVA APLICADAS EM OTIMIZAÇÃO RESTRITA	16
2.3 ABORDAGENS HÍBRIDAS APLICADAS EM OTIMIZAÇÃO RESTRITA	18
CAPÍTULO 3 ABORDAGEM NEURO-GENÉTICA PROPOSTA	21
3.1 ASPECTOS DE REDES NEURAIS ARTIFICIAIS.....	21
3.1.1 <i>A Rede Neural de Hopfield Convencional</i>	22
3.1.2 <i>Abordagem de Subespaço-Válido de Soluções</i>	27
3.1.3 <i>A Abordagem Neuro-Genética</i>	28
3.1.4 <i>Dinâmica da Rede de Hopfield Genética</i>	31
3.2 APLICAÇÃO DA REDE DE HOPFIELD GENÉTICA NA SOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO NÃO-LINEAR RESTRITA	35
3.3 CONSIDERAÇÕES FINAIS	41
CAPÍTULO 4 ASPECTOS DE ALGORITMOS GENÉTICOS UTILIZADOS NA ABORDAGEM PROPOSTA	43
4.1 EXPLORAÇÃO E PROSPECÇÃO	45
4.2 MÉTODOS E PARÂMETROS DO ALGORITMO GENÉTICO DESENVOLVIDO	46
4.2.1 <i>Representação Genética</i>	46
4.2.2 <i>População Inicial</i>	47
4.2.3 <i>Função de Avaliação</i>	48
4.2.4 <i>Métodos de Seleção</i>	48
4.2.5 <i>Operadores Genéticos</i>	49
4.2.6 <i>Critério de Parada</i>	51
4.3 CONSIDERAÇÕES FINAIS	51
CAPÍTULO 5 EXPERIMENTOS E RESULTADOS	53
5.1 EXPERIMENTOS BASEADOS EM REDES NEURAIS ARTIFICIAIS	54
5.1.1 <i>Experimento 1</i>	54
5.1.2 <i>Experimento 2</i>	58
5.1.3 <i>Experimento 3</i>	60
5.1.4 <i>Experimento 4</i>	63
5.1.5 <i>Experimento 5</i>	64
5.1.6 <i>Experimento 6</i>	66
5.1.7 <i>Experimento 7</i>	68
5.1.8 <i>Experimento 8</i>	71

5.1.9	<i>Experimento 9</i>	72
5.2	EXPERIMENTOS BASEADOS EM COMPUTAÇÃO EVOLUTIVA	74
5.3	PROBLEMAS COM REGIÕES DE BUSCA FACTÍVEIS DESCONEXAS	76
5.3.1	<i>Experimento 1</i>	77
5.3.2	<i>Experimento 2: Estudo de Caso em Identificação Robusta para Modelos Não-lineares</i>	77
5.4	CONSIDERAÇÕES FINAIS.....	82
CAPÍTULO 6 CONCLUSÕES E TRABALHOS FUTUROS		83
6.1	CONCLUSÕES.....	83
6.2	TRABALHOS FUTUROS.....	85
REFERÊNCIAS BIBLIOGRÁFICAS.....		87
APÊNDICE A IMPLEMENTAÇÃO EM HARDWARE DA REDE NEURAL DE HOPFIELD.....		94
APÊNDICE B ALGORITMO GENÉTICO COM CODIFICAÇÃO BINÁRIA		97

CAPÍTULO 1

Introdução

1.1 Objetivo e Motivação

Otimização é o processo de buscar a melhor solução (ou solução ótima) de um conjunto de soluções para um problema. A otimização pode ser dividida em duas classes: global e local, sendo que a otimização global encontra a melhor solução do conjunto entre todas as soluções possíveis e a otimização local encontra uma solução que não é a melhor de todas, mas se apresenta como a melhor dentro de um subconjunto do conjunto universo (Bazaraa et al., 1993).

Tipicamente, um problema de otimização possui três elementos constituintes: Variáveis de Decisão (parâmetros cujos valores definem uma solução para o problema), Função Objetivo (uma função das variáveis de decisão a ser minimizada ou maximizada) e Restrições (um conjunto de funções que define o espaço factível de soluções). Como exemplo, considere um sistema de produção. As variáveis de decisão podem representar as quantidades produzidas de determinados objetos; a função objetivo pode simbolizar o interesse em minimizar os custos na produção destes objetos; e as restrições podem estar relacionadas às limitações operacionais do processo de produção ou até mesmo limitações físicas e tecnológicas.

Quando a estrutura do problema é complexa ou existe uma grande variedade de possíveis soluções (como no caso do exemplo citado), geralmente não há uma solução

simples e diretamente calculável para o problema, havendo então a necessidade do uso de técnicas de otimização. Nesses casos, é improvável a existência de algum procedimento direto de solução, sendo que as técnicas de otimização podem ser assim utilizadas na busca pela melhor solução para o problema.

A aplicação de uma técnica de otimização depende da estrutura do problema. Se a função objetivo e as restrições são lineares, a programação linear é a melhor escolha (Bazaraa et al., 1993). Entretanto, o mundo real usualmente requer funções não-lineares e restrições de diferentes naturezas, havendo a necessidade de encontrar abordagens não apenas eficazes, mas também eficientes em resolver cada classe de problemas de otimização. Assim, com o intuito de melhorar o desempenho e a aplicabilidade em relação às abordagens já existentes, este trabalho é voltado ao desenvolvimento de uma abordagem híbrida, a qual envolve as metodologias de redes neurais artificiais e algoritmos genéticos, aplicada à otimização de diferentes tipos de problemas de característica não-linear.

Os sistemas baseados em redes neurais artificiais exploram a possibilidade de incorporar conhecimento a partir de dados disponíveis sobre determinado processo. Não são necessárias regras ou uma teoria que descreva o processo; as redes neurais simplesmente “aprendem” com os exemplos. Estes exemplos são apresentados sucessivamente à rede, que se adapta internamente a fim de representar o comportamento do processo. Por outro lado, os algoritmos genéticos, uma das vertentes da computação evolutiva, têm o objetivo de sintetizar em um programa de computador os processos de transmissão de material genético entre gerações de populações. Dado um conjunto inicial de possíveis soluções sub-ótimas (população), as mesmas podem ser combinadas (cruzamento de material genético) sucessivamente (gerações) até que um critério de parada seja satisfeito.

Desta forma, a finalidade dos projetos dessas arquiteturas híbridas visa tanto reduzir o tempo de convergência para soluções factíveis como também melhorar a precisão das

mesmas, aumentando assim, a eficiência global da arquitetura. Por conseguinte, a inserção de um algoritmo genético em uma arquitetura de rede neural, como esta desenvolvida aqui, evita as convergências para soluções consideradas infactíveis e pode ainda reduzir os respectivos tempos de convergência. Tal combinação neuro-genética tem também grande aceitação na comunidade científica, pois adere ao princípio de balanceamento de vantagens da inteligência computacional (Rezende, 2003; Jang et al., 1997), onde metodologias diferentes colaboram entre si potencializando a utilidade e aplicabilidade dos sistemas resultantes.

1.2 Justificativa e Relevância

Em relação à aplicação de redes neurais artificiais em otimização de sistemas, uma das primeiras abordagens foi proposta por Tank e Hopfield (1986), cuja idéia era resolver problemas de programação linear, a qual foi mapeada em um circuito eletrônico. Embora a rede de Tank e Hopfield tenha a desvantagem de seu ponto de equilíbrio nem sempre representar a solução do problema original, este trabalho pioneiro tem inspirado muitos pesquisadores a desenvolver outras redes neurais para resolver problemas de otimização linear e não-linear.

Kennedy e Chua (1988) estenderam e melhoraram o trabalho de Tank e Hopfield, desenvolvendo uma rede neural composta de um número finito de parâmetros de ponderação a fim de resolver problemas de otimização não-linear restrita. Por utilizar parâmetros de ponderação, a rede de Kennedy e Chua produz apenas soluções aproximadas, sendo que quando tais parâmetros forem inapropriadamente especificados, há então um elevado esforço computacional para obter soluções factíveis.

Com o objetivo de evitar o uso de parâmetros de ponderação, alguns estudos têm sido desenvolvidos desde a última década. Como exemplo, o trabalho de Rodríguez-Vázquez et al.

(1990) propôs uma rede neural para resolver uma classe de problemas de programação convexa não-linear. Esta rede é adequada para o caso em que as soluções iniciais estejam dentro da região factível do espaço de busca. Caso contrário, a rede pode não encontrar um ponto de equilíbrio (Zak et al., 1995). Já em Bouzerdoum e Pattison (1993), uma rede neural para resolver problemas de otimização convexa com variáveis canalizadas é também proposta, sendo que uma das restrições de tal abordagem é que a mesma somente pode ser utilizada em problemas com funções objetivo quadráticas.

Atualmente, utilizam-se arquiteturas de redes neurais diferentes, sistemáticas de mapeamento distintas ou a combinação de diferentes metodologias, com o intuito de mapear problemas de otimização não-linear. Algumas destas arquiteturas ainda apresentam dificuldades no mapeamento de problemas com restrições, resultando em soluções algumas vezes inviáveis. Assim, embora uma explanação mais abrangente em relação à aplicação de sistemas inteligentes em problemas de otimização não-linear restrita seja realizada no Capítulo 2, a partir das considerações efetuadas nos parágrafos anteriores se torna possível verificar que há ainda necessidade de investigar métodos alternativos que visem obter soluções factíveis, tendo como finalidade reduzir as particularidades estruturais impostas por grande parte das abordagens inteligentes apresentadas na literatura correlata.

1.3 Contribuições da Tese

A abordagem neuro-genética desenvolvida nesta tese supera em aplicabilidade e/ou eficiência as metodologias existentes para otimização não-linear restrita. Dentre as contribuições relacionadas à aplicabilidade, destacam-se as seguintes:

- (i) Mapeamento de problemas de otimização não-linear restrita de naturezas diferentes, utilizando uma mesma metodologia;

- (ii) Viabilização do tratamento das restrições envolvidas com os problemas;
- (iii) Independência em relação ao tipo de função objetivo ou tipo das restrições, tratando tanto de funções convexas ou não-convexas como também de equações de igualdade ou desigualdade;
- (iv) Ausência de necessidade de determinação, em cada iteração, do conjunto ativo de restrições;
- (v) Em cada iteração, não é preciso calcular uma direção admissível de busca;
- (vi) Implementação direta das variáveis canalizadas através da função de ativação dos neurônios da rede recorrente do tipo Hopfield;
- (vii) Desnecessidade de definição de parâmetros de controle ou ponderação, seja para inicialização ou durante a execução;
- (viii) A solução inicial não necessariamente precisa pertencer ao conjunto factível definido pelas restrições;
- (ix) Aplicável a problemas que possuem espaços de busca factíveis disjuntos;
- (x) Facilidade de implementação em hardware.

Em relação à eficiência, vale mencionar a simplicidade de implementação e de mapeamento de problemas da abordagem proposta, assim como a precisão das soluções obtidas. Em função da maioria dos trabalhos que tratam de problemas de otimização não-linear restrita não apresentar a informação sobre o tempo de convergência, torna-se difícil realizar qualquer análise neste sentido. Entretanto, análises comparativas referentes à precisão das soluções finais, assim como as particularidades envolvidas com a aplicação de outras arquiteturas alternativas utilizadas em otimização restrita, serão avaliadas tanto no Capítulo 2 como no Capítulo 5.

1.4 Organização do Trabalho

A redação do trabalho aqui proposto difere da forma tradicional de edição de teses e dissertações, a qual se caracteriza por primeiramente compor os capítulos referentes ao embasamento teórico, utilizado para a fundamentação do trabalho, seguido do capítulo que apresenta a proposta propriamente dita e, por fim, encerram-se com os capítulos de experimentos, resultados e conclusões.

Neste trabalho optou-se por desvincular a divisão dos capítulos entre aqueles com conteúdo teórico e aqueles com a proposta do trabalho. Deste modo, os conceitos são apresentados juntamente com a descrição de como foram aplicados na abordagem proposta. A justificativa principal para tal forma de redação é que a mesma facilita o entendimento seqüencial das fases envolvidas com o projeto estrutural da abordagem híbrida, baseada em redes neurais e em algoritmos genéticos, quando de sua aplicação na solução de problemas de otimização não-linear restrita.

A partir destas considerações, este trabalho está dividido em 6 capítulos, os quais são organizados como se segue.

No Capítulo 2 é apresentado um panorama de diversas abordagens inteligentes utilizadas na solução de problemas de otimização restrita, destacando em particular aquelas baseadas em redes neurais artificiais e em algoritmos genéticos.

No Capítulo 3 é apresentada a arquitetura neuro-genética desenvolvida, mostrando sua aplicação na solução de problemas de otimização não-linear restrita. São apresentados os conceitos fundamentais relativos às redes neurais artificiais, com enfoque principal sobre a rede recorrente utilizada, a qual incorpora a técnica do subespaço-válido de soluções para confinar as restrições envolvidas com os problemas.

O Capítulo 4 apresenta a estrutura, os componentes e os parâmetros do algoritmo genético utilizado na otimização da função objetivo dos problemas de otimização não-linear restrita.

Os experimentos com diferentes tipos de problemas de otimização não-linear e seus respectivos resultados são descritos no Capítulo 5.

Finalmente, no Capítulo 6, apresentam-se as conclusões e os trabalhos futuros.

Partes dos resultados já obtidos por intermédio do desenvolvimento desta tese já têm sido divulgados nos seguintes fóruns científicos:

BERTONI, Fabiana Cristina, SILVA, Ivan Nunes. “NeuroGenetic Approach for Solving Constrained Nonlinear Optimization Problems”. *The 13th International Conference on Neural Information Processing - Lecture Notes in Computer Science*, v. 4234, pp. 826-835, 2006.

BERTONI, Fabiana Cristina, SILVA, Ivan Nunes. “Uma Arquitetura Neuro-Genética para Otimização Não-Linear Restrita”. *Anais do XII Congresso Brasileiro de Automática*, v.1. pp.1938-1943, Salvador, Bahia, 2006.

BERTONI, Fabiana Cristina, SILVA, Ivan Nunes. “Efficient Neurogenetic Architecture for Solving General Constrained Optimization Problems”. *IEEE Transactions on Neural Networks (Submitted for Publication)*, 2007.

CAPÍTULO 2

Aspectos de Sistemas Inteligentes Aplicados em Otimização Não-Linear

Existem duas abordagens distintas para resolver um problema geral de otimização não-linear sujeito a restrições de diferentes naturezas (Zheng et al., 2002). A primeira utiliza métodos matemáticos tradicionais, os quais adotam técnicas de linearização, buscando formas de otimização para problemas de grande escala. Dentre os métodos mais utilizados, destacam-se os de Programação Quadrática Seqüencial (tais como o método de Lagrange, método de Newton e método quase-Newton) e os métodos de Função Barreira Modificada.

Os métodos de Programação Quadrática Seqüencial (*Sequential Quadratic Programming – SQP*) são métodos iterativos, nos quais para cada iteração se forma e se resolve um subproblema de otimização quadrática, considerando uma direção de busca e um tamanho de passo a ser dado ao longo desta direção. A eficiência destes métodos se torna baixa quando o problema de otimização apresenta um grande número de restrições e variáveis limitadas. Uma quantidade elevada de restrições ocasiona em um número maior de iterações necessárias para encontrar uma solução aceitável. A preocupação em garantir os limites de oscilação das variáveis também dificulta o processo de busca da solução.

Os métodos de Função Barreira Modificada convertem o problema restrito original em um novo problema sem restrições, encontrando dificuldades para essa transformação quando o problema possui restrições de igualdade. Para tentar solucionar este problema, os métodos

de Função Barreira Modificada transformam a restrição de igualdade em duas restrições de desigualdade equivalentes a ela, resolvendo-as da maneira usual. Entretanto, em aplicações reais de otimização, tais como processos industriais, restrições de igualdade aparecem com grande frequência e em elevada quantidade e, portanto, se cada uma for transformada em outras duas, a complexidade do problema é duplicada, o que torna o método impraticável (Zheng et al., 2002).

A segunda abordagem implementa soluções utilizando metodologias inteligentes, dentre as quais estão as redes neurais artificiais, a computação evolutiva e os sistemas fuzzy. Particularmente, as redes neurais têm se mostrado como uma das mais eficientes metodologias para otimização não-linear, em função de sua elevada taxa de computação e de sua facilidade de implementação em hardware (Zheng et al., 2002).

As seções a seguir apresentam alguns trabalhos que utilizam metodologias inteligentes na solução de problemas de otimização não-linear restrita. Tais trabalhos refletem o “Estado da Arte” das pesquisas relacionadas a este tema, contribuindo também para contextualizar a abordagem aqui proposta dentro do foco temático da área.

2.1 Abordagens de Redes Neurais Aplicadas em Otimização Restrita

Como mencionado no Capítulo 1, o trabalho de Bouzerdoum e Pattison (1993) apresentou resultados satisfatórios ao otimizar um problema não-linear com variáveis canalizadas, sendo restrito às funções objetivo quadráticas.

Em Silva (1997) é apresentada uma rede recorrente que fornece resultados satisfatórios para problemas de otimização restrita, mas a convergência em direção aos pontos de equilíbrio é lenta para a maioria das situações, sendo que este fato pode comprometer a

aplicabilidade efetiva da abordagem proposta. Outro aspecto importante é que a rede utiliza um parâmetro de inicialização obtido experimentalmente para cada problema em particular.

Alguns anos mais tarde, Liang e Wang (2000) publicaram um artigo propondo uma rede neural recorrente para otimização não-linear com função objetivo convexa e com variáveis canalizadas, sem a necessidade de que a função objetivo fosse do tipo quadrática. No entanto, para otimização de uma função objetivo quadrática, esta rede obtém bons resultados apenas se o ponto inicial da trajetória de solução estiver localizado dentro da região factível definida. Durante algumas simulações numéricas, foi observado que o tempo de convergência global da rede era elevado. Na tentativa de diminuir este tempo, percebeu-se que a rede se tornava mais sensível a erros. Algumas experiências com funções objetivo não-convexas foram realizadas, mas a rede convergia para mínimos locais. Baseado em algumas simulações com diferentes funções objetivo não-convexas, foi suposto que o problema estava relacionado com as características da função objetivo, não sendo realizada nenhuma análise mais detalhada do problema.

Em Xia et al. (2002) uma rede neural recorrente também demonstra ser estável para problemas de otimização convexa, mas assim como em Liang e Wang (2000), a rede possui um elevado tempo de convergência. Este problema ocorre em função de que a rede de Xia et al. (2002) utiliza parâmetros de ponderação das restrições, fazendo com que haja dispêndio de tempo na obtenção de valores ótimos para estes parâmetros.

Já o trabalho de Zheng et al. (2002) realiza otimização não-linear restrita através da associação de uma rede neural recorrente com o método do Multiplicador de Lagrange Aumentado, resultando em um novo algoritmo chamado pelos autores de Algoritmo ALM-NN (*Augmented Lagrange Multiplier – Neural Network*). O método do Multiplicador de Lagrange Aumentado, destinado à resolução de problemas de otimização não-linear restrita, é um método de Programação Quadrática Seqüencial que divide o problema original em

subproblemas quadráticos (diferenciáveis de segunda ordem) e tenta resolvê-los. Este método é falho ao tentar otimizar estes subproblemas, visto que sua função de penalidade é uma função diferenciável de primeira ordem apenas. O uso da rede neural vem suprir esta falha do método, utilizando a função de estabilidade de Lyapunov (duas vezes diferenciável) como função de penalidade do método de Lagrange Aumentado. Este novo algoritmo dispensa a utilização da matriz Hessiana (matriz de derivadas de segunda ordem da função), a qual demanda um maior consumo de memória e dificulta a convergência do sistema. No entanto, além de minimizar as variáveis envolvidas com o problema, a rede neural precisa minimizar também os pesos lagrangeanos e os parâmetros de penalidade envolvidos como o método de Lagrange Aumentado, o que ocasiona um tempo de processamento elevado, não apresentando melhorias significativas em relação aos trabalhos anteriores. Em relação à precisão das soluções encontradas, não é possível extrair conclusões significativas, uma vez que o trabalho apresenta apenas um único experimento, o qual obtém solução próxima à ótima.

Lua e Itob (2003) apresentam um método para converter problemas de otimização não-linear em problemas de otimização linear separáveis, usando redes neurais *feedforward*, tais como a rede Perceptron Multicamadas e a rede de Função de Base Radial. A idéia deste método é usar as habilidades das redes *feedforward* em expressar funções não-lineares em termos de composições de funções parametrizadas de uma única variável e em aproximar funções não-lineares contínuas arbitrárias com grau desejado de precisão. Após a aplicação da rede neural, funções lineares são obtidas, as quais são resolvidas pelo tradicional Método Simplex. O método proposto obteve apenas soluções aproximadas para todos os experimentos, o que já era esperado, uma vez que as funções lineares obtidas pela rede neural são aproximações de funções não-lineares.

Xia e Wang (2003) modificaram a rede neural recorrente proposta por Xia et al. (2002), de forma que esta não mais necessitasse da utilização de parâmetros de ponderação. A

rede resultante se mostra estável, simples de ser implementada e apresenta um tempo de convergência reduzido. No entanto, pode convergir para valores aproximados ou para ótimos locais.

Na continuidade de suas pesquisas, Xia e Wang (2004) desenvolveram um novo trabalho melhorando a precisão de seu modelo de rede recorrente apresentado em Xia e Wang (2003). As complexidades de mapeamento e implementação foram aumentadas, mas o novo modelo se mostrou capaz de tratar problemas de otimização convexa não-linear restrita, partindo de qualquer ponto do espaço de busca e convergindo para a solução ótima, com precisão de três casas decimais.

No ano de 2006, algumas publicações realizaram novas tentativas de melhorar o tempo de convergência de modelos que se valiam da simplicidade de utilizar parâmetros de ponderação para tratar as restrições juntamente com a função objetivo. O trabalho de Hao et al. (2006) utiliza uma rede neural de Hopfield para otimização não-linear restrita, utilizando multiplicadores de Lagrange para inserir as restrições na função objetivo. O trabalho apresenta bons resultados, mas relata que a eficiência computacional do modelo desenvolvido permanece baixa, mesmo com a utilização de processamento paralelo.

O trabalho desenvolvido por Ai et al. (2006) também utiliza uma rede neural de Hopfield associada a uma função de Lagrange, mas trata de problemas de otimização quadrática. Multiplicadores de Lagrange são utilizados para introduzir as restrições de igualdade e desigualdade separadamente na função objetivo. Em seguida, esta função de Lagrange é reconstruída, de forma a não introduzir variáveis de folga em função das restrições de desigualdade, o que permite que a rede tenha um número inferior de neurônios e um tempo de convergência reduzido quando comparada a outros modelos semelhantes da literatura. Nos experimentos apresentados, a rede sempre converge para o ponto de equilíbrio ótimo, partindo de qualquer ponto do espaço de busca.

Hu e Wang (2006) estenderam o trabalho desenvolvido em Xia e Wang (2004) para problemas não-convexos, uma vez que as redes neurais recorrentes existentes na literatura até o momento tratavam apenas de problemas de otimização convexos. O modelo desenvolvido também utiliza multiplicadores de Lagrange, inserindo as restrições na função objetivo, aplicando em seguida um método chamado *p-power* parcial, o qual tem a capacidade de localmente tornar a função Lagrangeana convexa em torno de um mínimo local (conversão do problema não convexo original em um problema convexo). A rede neural não converge para o ótimo global, mas fazer com que encontre um mínimo, mesmo que local, representa um grande avanço nas pesquisas em torno deste tema. Para que um problema não-convexo possa ser resolvido pela metodologia em questão, ele deve obedecer a critérios conhecidos como Condições de Karush-Kuhn-Tucker, os quais garantem que este problema pode ser transformado em um problema convexo equivalente. Portanto, a rede pode falhar para problemas que não obedeçam a esses critérios.

Xia e Feng (2007) propõem um novo modelo de rede neural de camada única para otimização não-linear restrita, a qual tem um comportamento semelhante a uma rede neural de projeção, desenvolvida em Xia et al. (2002), e a uma rede neural de Hopfield, mas é apresentada como mais rápida e precisa. O novo modelo de rede utiliza duas constantes de ponderação no mapeamento de problemas, as quais têm seus valores ótimos obtidos experimentalmente para cada problema. A rede se mostra mais precisa apenas quando estes valores ótimos são encontrados. A questão da convergência mais rápida também é questionável. A observação dos experimentos, tendo como parâmetros de análise o tempo de CPU e o número de iterações da rede, permite verificar uma diferença de aproximadamente 200 iterações entre o modelo de rede neural de projeção e o novo modelo de rede proposto, mas a diferença de tempo de CPU relativa a estas 200 iterações é de menos de 1 segundo.

Visando propósitos de sintetização das análises efetuadas nesta seção, a Tabela 2.1 mostra um resumo das abordagens neurais apresentadas, apontando suas principais ressalvas.

Tabela 2.1: Abordagens neurais aplicadas em problemas de otimização restrita.

Autores	Trabalho desenvolvido	Ressalvas
Silva (1997)	Rede recorrente para problemas de otimização restrita.	A convergência em direção aos pontos de equilíbrio é lenta. Utiliza um parâmetro de inicialização, dependente do problema, definido de forma experimental.
Liang e Wang (2000)	Rede neural recorrente para otimização não-linear com função objetivo convexa e com variáveis canalizadas.	Para funções quadráticas, o ponto inicial da trajetória de solução deve estar localizado dentro da região factível definida. Resultados insatisfatórios para funções objetivo não-convexas Tempo de convergência elevado.
Xia et al. (2002)	Rede neural recorrente para problemas de otimização convexa.	A rede possui um elevado tempo de convergência por utilizar parâmetros de ponderação.
Zheng et al. (2002)	Otimização não-linear restrita através da associação de uma rede neural recorrente com o método do Multiplicador de Lagrange Aumentado.	Tempo de processamento elevado.
Lua e Itob (2003)	Método para converter problemas de otimização não-linear em problemas de otimização linear separáveis usando redes neurais <i>feedforward</i> .	Obtém apenas soluções aproximadas.
Xia e Wang (2003)	Rede neural recorrente sem utilização de parâmetros de ponderação.	Convergência para valores aproximados ou para ótimos locais.
Xia e Wang (2004) (continuação do trabalho desenvolvido em 2003 pelos mesmos autores)	Rede neural recorrente sem utilização de parâmetros de ponderação.	Converge para a solução ótima, mas as complexidades de mapeamento e implementação são aumentadas.
Hao et al. (2006)	Rede neural de Hopfield para otimização não-linear restrita utilizando multiplicadores de Lagrange.	Baixa eficiência computacional, mesmo com a utilização de processamento paralelo.
Ai et al. (2006)	Rede neural de Hopfield associada a uma função de Lagrange.	Trata apenas de problemas de otimização quadrática.
Hu e Wang (2006)	Rede neural recorrente associada à técnica de multiplicadores de Lagrange para solução de problemas não-convexos.	Restrito a uma classe de problemas não-convexos. Não converge para o ótimo global.
Xia e Feng (2007)	Rede neural de camada única para otimização não-linear restrita.	Problemas de precisão e eficiência por utilizar parâmetros de ponderação.

2.2 Abordagens de Computação Evolutiva Aplicadas em Otimização Restrita

Técnicas de Computação Evolutiva possuem a tarefa de encontrar as soluções de um determinado problema dentre um conjunto de soluções em potencial, utilizando um mecanismo de busca em paralelo. O uso do paralelismo para a resolução de problemas pode ser uma boa opção, pois várias possibilidades são exploradas simultaneamente.

Alguns trabalhos procuram aplicar estas técnicas em problemas de otimização não-linear restrita. Koziel e Michalewicz (1999) propõem um algoritmo evolutivo que usa um decodificador para transformar um problema restrito em um problema irrestrito. Este algoritmo evolutivo utiliza um parâmetro adicional, dependente do problema, que deve ser obtido antes da execução do algoritmo, o qual divide o espaço de busca em subespaços. O algoritmo também requer esforço computacional adicional para encontrar todos os pontos que delimitam a região factível, definida pelas restrições. Por fim, o algoritmo apresenta dificuldades em resolver problemas não-convexos.

Deb (2000) usa um algoritmo genético associado a uma estratégia de penalidade, fazendo com que soluções factíveis tenham valores de *fitness* melhores que soluções infactíveis. O método é estável, mas falha quando os problemas possuem espaços de busca factíveis desconexos.

Os experimentos desenvolvidos por Ming et al. (2003) utilizam um Algoritmo Evolutivo com mecanismo de seleção de duas etapas: seleção de viabilidade (habilidade de sobrevivência) e seleção de fertilidade (habilidade de reprodução), executadas seqüencialmente durante o ciclo de vida dos indivíduos. Essas habilidades são avaliadas de acordo com as funções de penalidade e objetivo, respectivamente. Os parâmetros da função de penalidade são obtidos experimentalmente, havendo a necessidade de ajustá-los para cada problema.

Venkatraman e Yen (2005) desenvolveram um *framework* de duas fases, baseado em algoritmos genéticos. Na 1ª fase, a função objetivo é completamente desconsiderada e o problema de otimização restrita é tratado como um “problema de satisfação de restrições”. A busca genética tem o objetivo de minimizar a violação das restrições e encontrar uma solução viável. A solução com a menor violação de restrição é arquivada como a solução elite na população, sendo então preservada. Esta fase tem o critério da viabilidade de uma solução, tornando-se importante para problemas que apresentam um alto número de restrições. O algoritmo muda para a 2ª fase se pelo menos uma solução viável é identificada (as melhores soluções viáveis são salvas pelo Elitismo). Esta fase realiza a otimização da função objetivo e a satisfação das restrições simultaneamente, classificando-se como uma fase de otimização *bi*-objetivo. O *framework* proposto sempre encontra uma solução viável, que pode estar próxima da ótima ou não, se mostrando ineficiente em alguns casos.

O trabalho desenvolvido por Montes e Coelho (2005) apresenta uma Estratégia Evolutiva que não requer o uso de parâmetros de penalidade. Ao invés disso, a estratégia utiliza um mecanismo diversificado simples, permitindo que soluções inviáveis permaneçam na população. Um mecanismo de comparação é usado para guiar o processo em direção à região viável do espaço de busca e um tamanho de passo a ser dado ao longo desta direção é obtido experimentalmente. Os autores relatam problemas em encontrar um valor inicial para este tamanho de passo, colocando como pesquisa futura um estudo mais detalhado destes problemas. Em todos os experimentos, uma solução ótima ou bem próxima da ótima foi encontrada; no entanto, o custo computacional foi considerado elevado, tornando o método impraticável para problemas reais.

Em Wu et al. (2006) é proposto um algoritmo genético multi-população para resolver problemas de otimização restrita, em que cada população utiliza um valor de penalidade diferente, assim como a função de *fitness*. Deste modo, cada população evolui

independentemente com um determinado número de gerações, realizando posterior migração de indivíduos entre populações diferentes. Sendo assim, tal algoritmo pode realizar uma busca multi-direcional manipulando várias populações com seus respectivos graus de penalidade. O parâmetro de penalidade é obtido mais fácil e rapidamente, pois seu valor ótimo é procurado em diferentes populações de maneira simultânea. Nos experimentos realizados, o algoritmo apresentou respostas bem próximas à ótima.

A Tabela 2.2 apresenta uma síntese das abordagens evolutivas mencionadas, mostrando suas principais ressalvas.

Tabela 2.2: Abordagens evolutivas aplicadas em problemas de otimização restrita.

Autores	Trabalho desenvolvido	Ressalvas
Koziel e Michalewicz (1999)	Algoritmo evolutivo com decodificador.	Necessidade de parâmetro adicional dependente do problema. Elevado esforço computacional. Dificuldades em resolver problemas não-convexos.
Deb (2000)	Algoritmo genético associado a uma estratégia de penalidade.	O método falha quando os problemas possuem espaços de busca factíveis desconexos.
Ming et al. (2003)	Algoritmo evolutivo com mecanismo de seleção de duas etapas: viabilidade e fertilidade.	Eficiência reduzida por utilizar parâmetros de ponderação
Venkatraman e Yen (2005)	<i>Framework</i> de duas fases, baseado em algoritmos genéticos.	Nem sempre encontra uma solução ótima ou próxima da ótima.
Montes e Coelho (2005)	Estratégia evolutiva que não requer o uso de parâmetros de penalidade.	Passo a ser dado ao longo da direção de solução é obtido experimentalmente. Custo computacional elevado.
Wu et al. (2006)	Algoritmo genético multi-população, onde cada população utiliza um valor de penalidade diferente.	Tempo despendido para obter os valores ótimos dos parâmetros de penalidades.

2.3 Abordagens Híbridas Aplicadas em Otimização Restrita

A literatura apresenta alguns trabalhos que utilizam metodologias inteligentes combinadas, permitindo que estas colaborem entre si, suprindo falhas e, conseqüentemente, melhorando as soluções encontradas.

Le (1996) apresenta uma abordagem Fuzzy-Evolucionária para resolver problemas de otimização não-linear restrita. As restrições são associadas a pesos e introduzidas na função objetivo. Em seguida, um Algoritmo Evolutivo é utilizado para otimizar a função objetivo e um Sistema Fuzzy fica responsável por obter os valores dos parâmetros de ponderação que representam graus de satisfação das restrições. A abordagem obtém uma solução ótima aproximada, onde uma variável de erro representa um grau de tolerância de violação das restrições. O valor para esta variável é obtido experimentalmente.

Em Andino et al. (2000) é apresentado um método que integra técnicas de solução de restrições com um Algoritmo Evolutivo, utilizando uma linguagem de programação restrita para expressar o problema de forma declarativa, de modo que o método pode ser aplicado independente do problema. Durante os experimentos, verificou-se que o tempo de processamento do método era bastante elevado, o que motivou os autores a utilizarem processamento paralelo.

A Tabela 2.3 mostra um resumo das duas abordagens híbridas citadas, apresentando também suas principais ressalvas.

Tabela 2.3: Abordagens híbridas aplicadas em problemas de otimização restrita.

Autores	Trabalho desenvolvido	Ressalvas
Le (1996)	Abordagem fuzzy-evolucionária para resolver problemas de otimização não-linear restrita.	Solução ótima aproximada, com uma variável de erro obtida experimentalmente.
Andino et al. (2000)	Integra técnicas de solução de restrições com um algoritmo evolutivo.	Tempo de processamento bastante elevado.

O trabalho realizado na presente tese utiliza duas metodologias inteligentes, uma rede neural e um algoritmo genético, trabalhando de forma colaborativa na otimização de problemas restritos de característica não-linear. Entre as arquiteturas neurais descritas na Seção 2.1 e que poderiam ser utilizadas na abordagem neuro-genética proposta nesta tese, optou-se pela estrutura recorrente utilizada por Silva (1997), a qual foi inspirada a partir das

investigações introduzidas por Aiyer et al. (1990) a respeito do desempenho de redes de Hopfield quando aplicadas em problemas de otimização de sistemas. Entretanto, outras topologias recorrentes, tais como aquelas usadas por Hao et al. (2006) e Xia et al. (2002), poderiam ser também utilizadas para tal propósito. Já a proposta de incorporação do algoritmo genético dentro do sistema desenvolvido possui como meta fundamental a extração de suas potencialidades, que estão associadas à exploração de espaços de busca complexos e à flexibilidade no tratamento de funções não-lineares, sendo que estas habilidades são ausentes na maioria das redes neurais utilizadas em otimização não-linear restrita.

Assim, o que se busca nesta tese é a proposição de uma arquitetura híbrida neuro-genética que seja capaz de tratar problemas de otimização não-linear restrita de forma eficiente, contornando as limitações enfrentadas quando se utiliza uma única abordagem inteligente nesses tipos de problemas. Para tal propósito, apresenta-se no próximo capítulo a abordagem neuro-genética desenvolvida.

CAPÍTULO 3

Abordagem Neuro-Genética Proposta

3.1 Aspectos de Redes Neurais Artificiais

Sistemas baseados em Redes Neurais Artificiais (RNA) possuem elevadas taxas de processamento por utilizarem um grande número de elementos processadores simples, dispostos paralelamente e interconectados. Como as RNA também procuram representar a arquitetura do cérebro humano em hardware eletrônico, os sistemas baseados em RNA oferecem um método alternativo para solucionar problemas variados e complexos, tendo um potencial significativo para implementação em hardware (vide Apêndice A).

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. Tais modelos foram obtidos a partir da análise da geração e propagação de impulsos elétricos pela membrana celular dos neurônios (Hodgkin e Huxley, 1952).

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 por intermédio de sua função de saída (ativação) 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 (1943). 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 (Haykin, 1999).

Uma rede neural é definida por sua arquitetura e topologia, pelo tipo ou função do neurônio presente 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 de interligação entre neurônios, uma rede pode ser de Alimentação para Frente (*Feedforward*) ou Recorrente (Dinâmica). Em relação às redes de alimentação para frente, os neurônios são organizados em camadas e a informação se desloca em um único sentido, entre camadas adjacentes. Já nas redes recorrentes, não existe direção privilegiada para a propagação da informação, podendo haver realimentação. Um estudo abrangente sobre os vários tipos de redes neurais artificiais é apresentado em Haykin (1999) e Braga et al. (2000).

A rede desenvolvida neste trabalho é uma rede recorrente do tipo Hopfield, que utiliza a técnica denominada subespaço-válido de soluções para cálculo de seus parâmetros. 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.

3.1.1 A Rede Neural de Hopfield Convencional

Grande parte das abordagens neurais aplicadas em otimização de sistemas utilizam redes recorrentes, sendo que a rede neural de Hopfield é provavelmente o melhor exemplo

conhecido deste tipo de rede (Hush e Horne, 1993). Como definida em Hopfield (1984) e em Hopfield e Tank (1985), esta rede apresenta normalmente uma única camada com conexões realimentadas entre os nós (Figura 3.1). 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 neural de Hopfield contínua no tempo com N -neurônios é dada por:

$$\dot{u}_i(t) = \frac{\partial u_i}{\partial t} = -\eta u_i(t) + \sum_{j=1}^N T_{ij}(t) \cdot v_j(t) + i_i^b \quad (3.1)$$

$$v_i(t) = g_i(u_i(t)) \quad (3.2)$$

sendo que:

$\dot{u}_i(t)$ é a derivada de u_i em relação ao tempo (sistema dinâmico);

$u_i(t)$ 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.

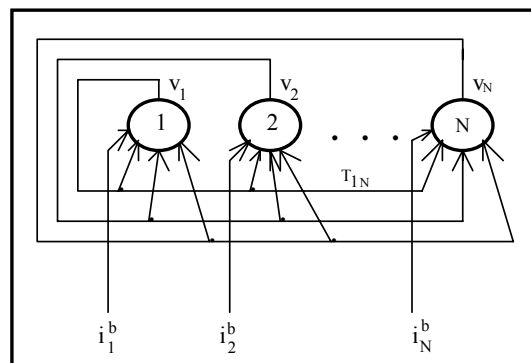


Figura 3.1. A rede neural de Hopfield convencional.

Desta forma, as respostas da rede de Hopfield são dinâmicas, isto é, aplica-se um conjunto de entradas; em seguida, as saídas v são calculadas e retro-alimentadas às entradas.

A saída é então recalculada e o processo é repetido iterativamente. Essas iterações produzem mudanças nas saídas cada vez menores, até que todas as saídas não mais apresentem alterações.

Na equação (3.2), $g_i(u_i(t))$ é uma função de ativação, monótona 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 as seguintes: função *threshold* (sinal), função *piecewise-linear* (rampa-simétrica) e função *sigmoid* (logística ou tangente hiperbólica). Uma versão discreta da rede neural de Hopfield é obtida quando $\eta=0$ e $T_{ij} = T_{ji}$ (Hopfield, 1984). Neste caso, os pontos de equilíbrio da rede correspondem aos valores de $v(t)$ que minimizam a 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:

$$E(t) = -\frac{1}{2} v(t)^T \cdot T \cdot v(t) - v(t)^T \cdot i^b \quad (3.3)$$

O mapeamento de problemas de otimização não-linear utilizando uma rede neural 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 (3.3). 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 neural de Hopfield se deve 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, mas também todas as restrições existentes. A rede atua com o propósito de minimizar simultaneamente uma função de energia $E^{ol}(t)$ correspondente à função objetivo e um 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 (Gee e Prager, 1991). Uma técnica simples de mapeamento é a que utiliza Multiplicadores de Lagrange (Bazaraa e Shetty, 1979), a qual consiste em incluir as restrições como termos na função de energia que são minimizados quando as restrições são satisfeitas, ou seja:

$$E(t) = E^{ot}(t) + c_1.E_1^{rest}(t) + c_2.E_2^{rest} + \dots + c_k.E_k^{rest} \quad (3.4)$$

sendo que 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 (1985). Em tal trabalho são descritos os resultados obtidos, em simulações por computador, da rede neural 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 as seguintes:

- (i) 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 (Hegde et al., 1988; Kamgar-Parsi e Kamgar-Parsi, 1990).
- (ii) Influência dos termos de restrições no termo de otimização, dificultando a convergência da rede (Peterson e Soderberg, 1989; Wilson e Pawley, 1988).
- (iii) A qualidade da solução final é afetada pelos valores das constantes de ponderação.

Em resumo, no processo de otimização com uma rede neural de Hopfield, cuja função objetivo é dada por (3.4), 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 .

Para contornar estes problemas, realiza-se em Aiyer et al. (1990) uma análise dos autovetores da rede durante a sua convergência, o que permite verificar como $E^{ot}(t)$ e os termos de restrições E_k^{rest} em (3.4) podem ser efetivamente separados dentro de subespaços diferentes, de modo a tornar o problema factível. O subespaço que agrupa todas as restrições impostas pelo problema é denominado subespaço-válido de soluções (Seção 3.1.2). Neste caso, a equação definida por (3.4) é simplificada para:

$$E(t) = E^{ot}(t) + c_0.E^{conf}(t) \quad (3.5)$$

sendo que o termo $E^{conf}(t)$ satisfaz todas as restrições E_k^{rest} dadas na equação (3.4).

Dado que $E(t)$ foi dividido em um termo de confinamento ($E^{conf}(t)$) e em um termo de otimização ($E^{ot}(t)$) que pode ser representativo da função custo do problema, tal que a minimização de $E^{conf}(t)$ da rede confina v dentro de um subespaço-válido e a minimização de $E^{ot}(t)$ da rede move v em direção a uma solução ótima, então, similarmente à equação (3.3), os termos da função de energia em (3.5) podem ser escritos como:

$$E^{ot}(t) = -\frac{1}{2} v(t)^T . T^{ot} . v(t) - v(t)^T . i^{ot} \quad (3.6)$$

$$E^{conf}(t) = -\frac{1}{2} v(t)^T . T^{conf} . v(t) - v(t)^T . i^{conf} \quad (3.7)$$

Devido à necessidade de assegurar que $E^{conf}(t)$ seja dominante (satisfação das restrições) sobre $E^{ot}(t)$, deve-se atribuir um valor elevado para a constante c_0 em (3.5). Esta condição torna a simulação da rede ineficiente, visto que a maioria do tempo será despendida no confinamento de v dentro do subespaço-válido. Para evitar este problema, propõe-se na Seção 3.1.2 uma estratégia de obtenção dos parâmetros do termo $E^{conf}(t)$ que diretamente garante a validade das restrições agrupadas pelo mesmo, dispensando a utilização da constante de ponderação c_0 .

3.1.2 Abordagem de Subespaço-Válido de Soluções

Para uma grande variedade de problemas com restrições que podem ser resolvidos pela rede neural de Hopfield, através da decomposição da função de energia da rede (3.3) nas duas parcelas dadas em (3.6) e (3.7), verifica-se que os pontos de equilíbrio da rede, correspondentes aos valores de v que minimizam a função de energia E^{conf} dada pela equação (3.7), pertencem todos a um mesmo subespaço-comum (Aiyer et al., 1990). Este subespaço comum, denominado subespaço-válido de soluções, possui equação definida por:

$$v(t+1) = T^{val}.v(t) + s \quad (3.8)$$

sendo que:

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

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

- (i) A rede é inicializada em um estado aleatório tal que v seja limitado pela função de ativação dos neurônios;
- (ii) A minimização de $E^{conf}(t)$ por intermédio da projeção de v no subespaço-válido, move v em direção a um dos pontos de equilíbrio da rede que satisfaz todas as restrições impostas pelo 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 Silva (1995).

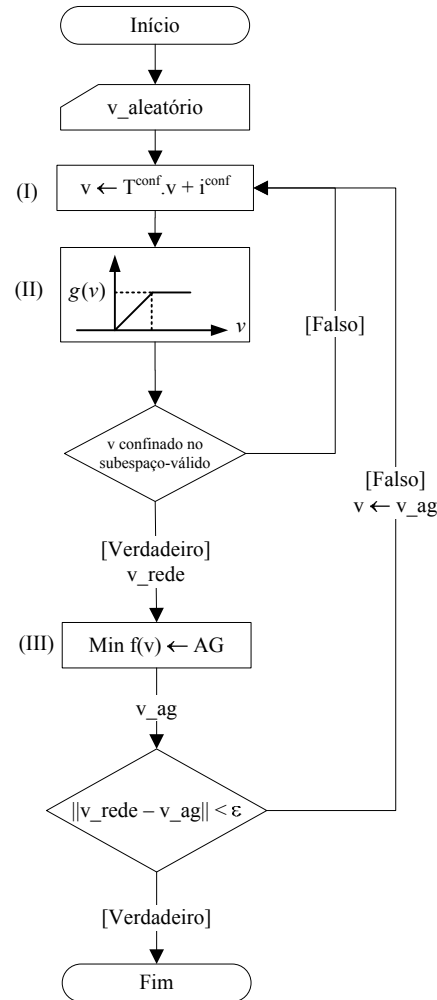
3.1.3 A Abordagem Neuro-Genética

Analogamente à Equação (3.5), a função de energia $E(t)$ utilizada neste trabalho será composta de dois termos:

$$E(t) = \text{Min } f(v) + E^{\text{conf}}(t) \quad (3.9)$$

sendo que $E^{\text{conf}}(t)$ é um termo de confinamento que agrupa as restrições estruturais impostas pelo problema, e $\text{Min } f(v)$ é a minimização da função objetivo do problema em questão, a qual conduz a saída da rede para os pontos de equilíbrio que representam as soluções do problema de otimização considerado.

Como descrito anteriormente, o modelo convencional da rede de Hopfield necessita de um esforço computacional adicional para forçar o confinamento de v no subespaço-válido, gastando tempo para isso. Com o propósito de contornar tais problemas, propõe-se a implementação de uma Rede de Hopfield Genética (RHG), apresentada na Figura 3.2.



$v_ag = v$ proveniente do algoritmo genético
 $v_rede = v$ proveniente da rede neural de Hopfield

Figura 3.2. A Rede de Hopfield Genética.

A dinâmica desta rede pode ser explicitada através dos três passos que compõem o esquema da Figura 3.2:

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

$$v(t+1) = T^{conf}.v(t) + i^{conf} \quad (3.10)$$

Dado que T^{conf} é uma matriz projeção, qualquer componente de v que seja ortogonal a T^{conf} , v é desprezada (Seção 3.1.4). Esta operação realiza uma minimização indireta de E^{conf} , sendo que $T^{conf} = T^{val}$ e $i^{conf} = s$.

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

$$g_i(v_i) = \begin{cases} \lim_i^{\inf} & \text{se } \lim_i^{\inf} > v_i \\ v_i & \text{se } \lim_i^{\inf} \leq v_i \leq \lim_i^{\sup} \\ \lim_i^{\sup} & \text{se } v_i > \lim_i^{\sup} \end{cases} \quad (3.11)$$

sendo que $v_i \in [\lim_i^{\inf}, \lim_i^{\sup}]$. Este passo assegura que os elementos do vetor v de saída da rede estejam sempre entre $[\lim_i^{\inf}, \lim_i^{\sup}]$.

Após a convergência do loop interno, o qual é realizado através das aplicações sucessivas dos passos (I) e (II), a RHG está pronta para executar o terceiro passo de otimização, ou seja:

(III) Minimização de $f(v)$: Com base nos valores obtidos para v , após seu confinamento para um subespaço-válido, aplica-se um algoritmo genético para a alteração de v em direção a uma solução ótima frente à função objetivo do problema, a qual corresponderá à minimização de $E^{ot}(t)$.

Como observado na Figura 3.2, as aplicações sucessivas dos passos (I) e (II), seguidas pela execução do passo (III), levam a saída da RHG para um ponto de equilíbrio que corresponde à solução ótima do problema de otimização restrita. Neste caso, a iteração representada pelos passos (I), (II) e (III) mencionados tem dois estágios distintos. No primeiro estágio, v é projetado diretamente no subespaço-válido a fim de satisfazer as restrições impostas para o problema. Este primeiro estágio é um processo iterativo, em que v é ortogonalmente projetado no subespaço-válido por meio da aplicação do Passo (I), seguido pela aplicação do Passo (II), a fim de limitar seus elementos no domínio definido por

$[\lim_i^{\text{inf}}, \lim_i^{\text{sup}}]$. No segundo estágio, aplicado após a minimização efetuada nos Passos (I) e (II), v é alterado em direção a uma solução ótima para o problema utilizando um algoritmo genético (Passo (III)).

A aplicação dos passos (I), (II) e (III) é repetida sucessivamente, enquanto os valores de v devolvidos pela rede de Hopfield (v^{rede}) e pelo algoritmo genético ($v^{\text{genético}}$) não forem bem similares. O processo de convergência é concluído quando os valores de v^{rede} e $v^{\text{genético}}$ estejam bem próximo um do outro entre duas iterações sucessivas, considerando para tanto a tolerância (precisão) requerida para a comparação dos mesmos.

3.1.4 Dinâmica da Rede de Hopfield Genética

Nesta seção, analisa-se a convergência da Rede de Hopfield Genética cuja dinâmica de operação é implementada através dos passos descritos na Figura 3.2. 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’ (3.11). A equação nodal descrevendo o comportamento dinâmico desta rede é obtida a partir de (3.1) para $\eta = 0$ e $v(t) = u(t)$, ou seja:

$$\dot{v}_i(t) = \frac{\partial v_i}{\partial t} = \sum_{j=1}^N T_{ij} \cdot v_j(t) + I_i^b \quad (3.12)$$

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 Genética, v é constantemente confinado ao subespaço-válido pela aplicação da operação (I), isto é $v = T^{\text{val}} \cdot v + s$, 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, ou seja:

$$\begin{aligned}
\dot{v}^{val} &= T^{val} \cdot \dot{v} = T^{val}(T^{ot} \cdot v + i^{ot}) \\
&= T^{val}(T^{ot}(T^{val} \cdot v + s) + i^{ot}) \\
&= T^{val}T^{ot}T^{val} \cdot v + T^{val}(T^{ot} \cdot s + i^{ot})
\end{aligned} \tag{3.13}$$

Da equação (3.13), verifica-se que \dot{v}^{val} é composta de duas partes: um termo independente de v , $T^{val}(T^{ot} \cdot s + i^{ot})$, e um termo que depende de v , $T^{val}T^{ot}T^{val}v$. Para simplificar, estes termos passam a ser definidos por:

$$T^{val}T^{ot}T^{val} = A \tag{3.14}$$

$$T^{val}(T^{ot} \cdot s + i^{ot}) = b \tag{3.15}$$

Assim, a equação (3.13) torna-se:

$$\dot{v}^{val} = A \cdot v + b = A \cdot v^{val} + b \tag{3.16}$$

sendo que $v^{val} = T^{val} \cdot v$ e $T^{val} \cdot T^{val} = T^{val}$.

Para sistemas invariantes no tempo (autônomos), a solução geral de (3.16) pode ser descrita por meio de uma equação exponencial matricial (D'Azzo e Houpis, 1975; Rosenbrook e Storey, 1970; Vidyasagar, 1993b):

$$v^{val}(t) = e^{At} v_0^{val} + \int_0^t e^{A(t-\tau)} b \cdot d\tau \tag{3.17}$$

sendo que v_0^{val} é o valor de 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 $e^{A\zeta}$ é dada por:

$$e^{A\zeta} = \sum_{k=0}^{\infty} \frac{(A\zeta)^k}{k!} \tag{3.18}$$

a equação (3.17) torna-se:

$$\begin{aligned}
v^{val}(t) &= \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k v_0^{val} + \int_0^t \sum_{k=0}^{\infty} \frac{(t-\tau)^k}{k!} A^k b \cdot d\tau \\
&= \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k v_0^{val} + \sum_{k=0}^{\infty} \frac{A^k b}{k!} \int_0^t (t-\tau)^k \cdot d\tau \\
&= \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k v_0^{val} + \sum_{k=0}^{\infty} \frac{A^k b}{k!} \left(\frac{t^{k+1}}{k+1} \right) \\
&= \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k v_0^{val} + \sum_{k=0}^{\infty} \frac{t^{k+1}}{(k+1)!} A^k b
\end{aligned} \tag{3.19}$$

Para se analisar o comportamento de v^{val} durante o processo de convergência da rede, deve-se escrever os vetores v^{val} , v_0^{val} e b em termos de suas componentes expressas no espaço coordenado gerado pelos autovetores normalizados da matriz A . Para isto, considere $\lambda_1, \lambda_2, \dots, \lambda_N$ os autovalores de A , com autovetores normalizados u_1, u_2, \dots, u_N . Para se distinguir entre os autovalores nulos e não-nulos de 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 v^{val} , v_0^{val} e b na direção dos autovetores de A gera:

$$v^{val} = \sum_{i=1}^N v_i u_i, \quad v_0^{val} = \sum_{i=1}^N o_i u_i, \quad b = \sum_{i=1}^N b_i u_i \tag{3.20}$$

sendo que v_i , o_i e b_i são, respectivamente, os valores da i -ésima componente dos vetores v^{val} , v_0^{val} e b , representadas no espaço coordenado gerado pelos autovetores de A .

A partir de (3.20), obtém-se:

$$A^k v_0^{val} = \sum_{i=1}^N o_i \lambda_i^k u_i, \quad A^k b = \sum_{i=1}^N b_i \lambda_i^k u_i \tag{3.21}$$

Com a utilização de (3.21), a equação (3.19) torna-se:

$$\begin{aligned}
v^{val}(t) &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \sum_{i=1}^N o_i \lambda_i^k u_i + \sum_{k=0}^{\infty} \frac{t^{k+1}}{(k+1)!} \sum_{i=1}^N b_i \lambda_i^k u_i \\
&= \sum_{i=1}^N o_i u_i \underbrace{\sum_{k=0}^{\infty} \frac{t^k \lambda_i^k}{k!}}_{e^{\lambda_i t}} + \sum_{i \notin Z} \frac{b_i u_i}{\lambda_i} \sum_{k=0}^{\infty} \frac{t^{k+1} \lambda_i^{k+1}}{(k+1)!} + \sum_{i \in Z} b_i u_i \sum_{k=0}^{\infty} \frac{t^{k+1} 0^k}{(k+1)!} \\
&= \sum_{i=1}^N e^{\lambda_i t} o_i u_i + \sum_{i \notin Z} \frac{b_i u_i}{\lambda_i} \left(\underbrace{\sum_{k=0}^{\infty} \frac{t^k \lambda_i^k}{k!}}_{e^{\lambda_i t}} - 1 \right) + \sum_{i \in Z} b_i u_i t \\
&= \sum_{i=1}^N e^{\lambda_i t} o_i u_i + \sum_{i \notin Z} \frac{b_i u_i}{\lambda_i} (e^{\lambda_i t} - 1) + \sum_{i \in Z} b_i u_i t
\end{aligned} \tag{3.22}$$

A equação (3.22) é válida para valores arbitrários de A , b e v_0^{val} . Entretanto, pode-se simplificar esta equação a partir das definições de A e b dadas nas equações (3.14) e (3.15). Desta análise, verifica-se que b permanece integralmente no subespaço-válido durante toda convergência, de modo que $b_i = 0$ para $i \in Z$. Assim, a equação (3.22) torna-se:

$$v^{val}(t) = \sum_{i=1}^N e^{\lambda_i t} o_i u_i + \sum_{i \notin Z} \frac{b_i u_i}{\lambda_i} (e^{\lambda_i t} - 1) \tag{3.23}$$

Para t pequeno, tem-se que $e^{\lambda_i t} \approx 1 + \lambda_i t$. Logo, a equação (3.23) torna-se:

$$v^{val}(t) \approx \sum_{i=1}^N [o_i (1 + \lambda_i t) + b_i t] u_i \tag{3.24}$$

Como v_0^{val} é escolhido aleatoriamente pequeno, os termos o_i são também pequenos em comparação com os b_i . Portanto, a equação (3.24) transforma-se em:

$$v^{val}(t) \approx t \cdot \sum_{i=1}^N b_i u_i = t \cdot b \tag{3.25}$$

Verifica-se então que os valores de v^{val} inicialmente partem na direção do vetor b . No limite, para t grande, e visto que v é limitado pelo hipercubo definido pela função de ativação rampa-simétrica dada em (3.11), a equação (3.23) indica que v^{val} tenderá à direção dos autovetores de A correspondente ao maior autovalor positivo, sendo que u^{max} denota estes

autovetores (e λ^{max} o autovalor correspondente) (Vidyasagar, 1993b). 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 Genética, inicializada em um ponto aleatório, convergirá para um ponto de equilíbrio válido. Este ponto de equilíbrio está contido dentro do hipercubo definido pela função de ativação ‘rampa-simétrica’.

A próxima seção introduz alguns conceitos fundamentais relacionados à otimização não-linear, os quais serão utilizados neste trabalho. Tais conceitos auxiliarão na análise das condições necessárias para a convergência da rede em direção aos pontos de equilíbrio. Em seguida, é apresentada a metodologia utilizada no mapeamento de problemas de otimização não-linear restrita através da Rede de Hopfield Genética.

3.2 Aplicação da Rede de Hopfield Genética na Solução de Problemas de Otimização Não-Linear Restrita

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 (Bazaraa e Shetty, 1979). O termo “problema de otimização restrita” refere-se à ação de minimizar ou maximizar uma função objetivo na presença destas restrições. Se nenhuma restrição estiver associada à função objetivo, o problema é dito ser um “problema de otimização irrestrita”.

Assim, seja um conjunto de restrições $h(x)$, formado por funções contínuas e diferenciáveis, definidas por:

$$\begin{aligned}
h_1(x) &= 0 \\
h_2(x) &= 0 \\
&\dots \\
h_p(x) &= 0
\end{aligned} \tag{3.26}$$

A partir da teoria de sistemas não-lineares, têm-se os seguintes teoremas e definições:

Definição 3.1 (Luenberger, 1984): Um ponto regular x^* satisfazendo as restrições $h(x^*)$ é considerado um ponto regular das restrições se os valores dos gradientes $\nabla h_1(x^*)$, $\nabla h_2(x^*)$, ..., $\nabla h_p(x^*)$ forem linearmente independentes, sendo que a matriz gradiente $\nabla h(x)$ é definida por:

$$\nabla h(x) = \begin{bmatrix} \frac{\partial h_1(x)}{\partial x_1} & \frac{\partial h_1(x)}{\partial x_2} & \dots & \frac{\partial h_1(x)}{\partial x_N} \\ \frac{\partial h_2(x)}{\partial x_1} & \frac{\partial h_2(x)}{\partial x_2} & \dots & \frac{\partial h_2(x)}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_p(x)}{\partial x_1} & \frac{\partial h_p(x)}{\partial x_2} & \dots & \frac{\partial h_p(x)}{\partial x_N} \end{bmatrix} = \begin{bmatrix} \partial h_1(x)^T \\ \partial h_2(x)^T \\ \vdots \\ \partial h_p(x)^T \end{bmatrix} \tag{3.27}$$

Definição 3.2 (Bazaraa e Shetty, 1979): Um conjunto $C \subseteq R^N$ é considerado convexo se para todo $x_1, x_2 \in C$, o ponto $\lambda x_1 + (1 - \lambda)x_2 \in C$, sendo que $\lambda \in [0, 1]$.

Definição 3.3 (Bazaraa e Shetty, 1979): Um ponto x pertencente a um conjunto convexo C é considerado um ponto extremo de C se não existem pontos $x_1, x_2 \in C$, tal que $x = \lambda x_1 + (1 - \lambda)x_2$, sendo que $\lambda \in [0, 1]$.

Definição 3.4 (Bazaraa e Shetty, 1979): Toda matriz de projeção é simétrica e semidefinida positiva.

Teorema 3.1 (Luenberger, 1984): Num ponto regular x^* da superfície S definida por $h(x) = 0$, o plano tangente M é definido por $M = \{x: \nabla h(x^*)x = 0\}$.

Teorema 3.2 (Bazaraa e Shetty, 1979): Dado que C é um conjunto convexo em R^N , a minimização de uma função $f(x)$, sujeita a $x \in C$, tem uma solução ótima global se $f(x)$ for convexa.

Teorema 3.3 (Bazaraa e Shetty, 1979): Seja $f(x)$ uma função duas vezes diferenciável num ponto x^* . Se x^* é um ponto de mínimo, então $\nabla f(x^*) = 0$, e a matriz Hessiana $H(x^*)$, cujos elementos são as derivadas parciais de segunda ordem de $f(x^*)$, é semidefinida positiva.

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

$$\begin{aligned} g_1(x) &\leq 0 \\ g_2(x) &\leq 0 \\ &\dots \\ g_q(x) &\leq 0 \end{aligned} \tag{3.28}$$

Na próxima subseção apresenta-se a metodologia de mapeamento de otimização restrita através da Rede de Hopfield Genética.

Metodologia de Mapeamento de Problemas através da RHG

Problemas de otimização não-linear restrita com N -variáveis de decisão, tendo p -restrições de igualdade e q -restrições de desigualdade, podem ser descritos pelas seguintes equações:

$$\text{Minimizar } f(v) \tag{3.29}$$

$$\text{Sujeito a: } \begin{cases} h_i(v) = 0, i \in \{1..p\} \end{cases} \tag{3.30}$$

$$\begin{cases} g_j(v) \leq 0, j \in \{1..q\} \end{cases} \tag{3.31}$$

$$z^{mim} \leq v \leq z^{max} \tag{3.32}$$

sendo que $v, z^{mim}, z^{max} \in R^N$; $g(v) \leq 0$ e $h(v) = 0$ definem um conjunto convexo em R^N , e as funções $g_i(v)$ e $h_j(v)$ são funções contínuas e diferenciáveis. As equações analíticas para os

parâmetros internos da rede, a serem definidos por T^{conf} e i^{conf} em (3.10), são explicitadas a seguir.

Considera-se, inicialmente, o caso em que existam apenas restrições de igualdade na forma descrita pela equação definida em (3.30). Para sistemas não-lineares, torna-se difícil estabelecer alguma suposição sobre qualquer solução admissível inicial, que resulte na derivação de um subespaço-válido de soluções. Entretanto, com base na teoria de estabilidade de Lyapunov (Vidyasagar, 1993b), 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 pode ser geralmente representada por um sistema de equações diferenciais de primeira ordem com a seguinte forma (Rosenbrook e Storey, 1970):

$$\dot{x} = \frac{dx(t)}{dt} = F(x(t)) \quad (3.33)$$

sendo que a função $F(x(t))$ representa um conjunto de equações (em geral) não-lineares, na forma dada pela Equação (3.30). Assim, considerando um sistema dinâmico descrito pela equação (3.33), um vetor x^e é considerado um estado de equilíbrio do sistema se for satisfeita a seguinte equação (Fang e Kincaid, 1996; Vidyasagar, 1993a):

$$F(x^e) = 0 \quad (3.34)$$

sendo que $F(\cdot)$ é comumente diferenciável.

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

$$F(x) \cong F(x^e) + A.(x - x^e) \quad (3.35)$$

sendo que a matriz A é definida como o Jacobiano de $F(x)$ em relação a x , ou seja, $A = \nabla F(x)$.

Assim, analisando-se o valor de $F(x)$ na vizinhança do estado $x^e = 0$, obtém-se a condição de equilíbrio dada em (3.34), ou seja:

$$\lim_{x \rightarrow x^e} \frac{\|F(x)\|}{\|x\|} = 0 \quad (3.36)$$

sendo que o símbolo $\|\cdot\|$ denota a norma Euclidiana. Logo, levando-se em consideração esta condição, a Equação (3.35) torna-se:

$$F(x) \cong A.x = \nabla F(x).x \quad (3.37)$$

É importante notar que a equação (3.37) é válida, sem perda de generalidade, para qualquer ponto de equilíbrio $x^e \in R^N$. Para um ponto de equilíbrio $x^e \neq 0$, podem-se transladar os eixos coordenados a fim de que a origem do novo sistema seja coincidente com o x^e (Fang e Kincaid, 1996). Baseado nestes conceitos, deduz-se, então, que uma solução para o sistema de equações não-lineares, dado por (3.30), é o próprio vetor $v^e = 0$, que será introduzido na equação do subespaço-válido através do vetor s .

Por outro lado, a matriz projeção T^{val} da equação do subespaço-válido é obtida através da projeção de v para o subespaço tangente à superfície delimitada pelas restrições dadas pelas equações (3.30) e (3.31) (Bazaraa e Shetty, 1979). Assim, uma equação para T^{val} pode ser definida a partir da equação linearizada (3.37), por:

$$T^{val} = I - \nabla h(v)^T \cdot (\nabla h(v) \cdot \nabla h(v)^T)^{-1} \cdot \nabla h(v) \quad (3.38)$$

Substituindo-se a matriz T^{val} na equação do subespaço-válido, obtém-se a seguinte expressão iterativa:

$$v \leftarrow [I - \nabla h(v)^T \cdot (\nabla h(v) \cdot \nabla h(v)^T)^{-1} \cdot \nabla h(v)].v + s \quad (3.39)$$

Por fim, as condições deduzidas a partir de (3.36) e (3.37) devem ser introduzidas na Equação (3.39) 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 a Equação (3.39), tem-se:

$$v \leftarrow I.v - \nabla h(v)^T \cdot (\nabla h(v) \cdot \nabla h(v)^T)^{-1} \cdot \nabla h(v) \cdot v + s \quad (3.40)$$

Utilizando-se o resultado obtido em (3.37), conclui-se que quando v ($\|s\| \rightarrow 0$) tende ao equilíbrio, a Equação (3.40) torna-se:

$$v \leftarrow v - \nabla h(v)^T \cdot (\nabla h(v) \cdot \nabla h(v)^T)^{-1} \cdot h(v) \quad (3.41)$$

Logo, a equação (3.41) sintetiza a equação do subespaço-válido para sistemas de equações não-lineares. Neste caso, a equação original do subespaço-válido em (3.10), representando a minimização indireta de E^{conf} , deve ser substituída pela Equação (3.41). A aplicação sucessiva do passo (I) seguido do passo (II) da Figura 3.2, faz com que v seja uma solução que satisfaça todas as restrições impostas pelo problema de otimização não-linear restrita.

A metodologia utilizada para as restrições de igualdade não-linear pode também ser estendida para as restrições de desigualdade não-linear dada pela Equação (3.31). Logo, uma restrição de desigualdade típica dada por $g_i(v) \leq 0$, torna-se:

$$g_i(v) + \sum_{j=1}^q q_j \cdot w_j = 0 \quad (3.42)$$

sendo que w_j são variáveis auxiliares (tratadas como variáveis do vetor v), e q_j são constantes definidas pela função impulso de Kronecker:

$$q_j = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases} \quad (3.43)$$

Assim, neste caso, transforma-se um conjunto de restrições de desigualdade em um conjunto de restrições de igualdade.

3.3 Considerações Finais

Com o objetivo de fornecer uma nova metodologia para mapeamento de problemas de otimização não-linear e de aperfeiçoar a eficiência de simulações por computador, uma arquitetura neuro-genética foi apresentada neste capítulo. Trata-se de uma arquitetura híbrida, constituída por uma rede neural de Hopfield modificada e por um algoritmo genético.

A Rede de Hopfield Genética utiliza a técnica do subespaço-válido de soluções para realizar a minimização da função de energia da rede. A técnica de subespaço-válido utilizada 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 solucionado. Esta técnica se baseia na premissa de que todos os vetores de saída da rede de Hopfield, que correspondem às soluções válidas de um problema específico, pertencem todos a um mesmo subespaço.

Assim, a operação da Rede de Hopfield Genética é executada através de três passos principais, definidos por:

- (I) Projeção de v no subespaço-válido (minimização indireta de E^{conf}).
- (II) Aplicação da função de ativação “rampa-simétrica”. A partir destes dois passos iniciais, e utilizando ferramentas de análise de sistemas dinâmicos, foi possível estudar o comportamento da rede, bem como a sua convergência para os pontos de equilíbrio. Foi mostrada que estes pontos de equilíbrio são sempre uma solução válida para o problema em análise.

(III) O algoritmo genético realiza um terceiro passo, que corresponde à minimização da função objetivo do problema. Após a minimização de E^{conf} pelos passos (I) e (II), os valores de v são repassados ao algoritmo genético que os utiliza como uma possível solução para o problema de otimização.

Por fim, uma breve introdução à otimização não-linear foi também apresentada e uma metodologia de mapeamento de problemas de otimização restrita utilizando a Rede de Hopfield Genética foi proposta.

CAPÍTULO 4

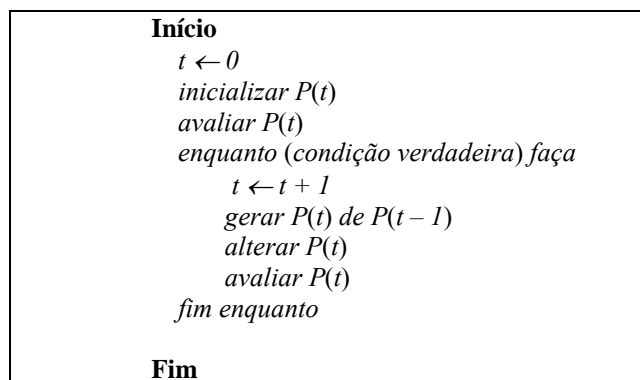
Aspectos de Algoritmos Genéticos Utilizados na Abordagem Proposta

Muitos problemas computacionais requerem um processo de busca de soluções otimizadas a partir da consideração de inúmeras possíveis soluções disponíveis no espaço de busca. O termo *espaço de busca* se refere a uma coleção de soluções candidatas para um determinado problema (Mitchell, 1996). Os Algoritmos Genéticos (AG), introduzidos por Holland (1975), são algoritmos de busca baseados nos mecanismos da seleção natural e da genética. A idéia que envolve os AG é a de simular processos naturais de sobrevivência e reprodução dos seres vivos que são essenciais em sua evolução. Na natureza, os indivíduos de uma mesma população competem entre si, buscando principalmente a sobrevivência, e, quanto melhor um indivíduo se adaptar ao meio em que vive, maior será sua chance de sobreviver e gerar descendentes.

Para reproduzir estes processos naturais, os algoritmos genéticos retiram cromossomos (cadeias de símbolos) de uma população de cromossomos e os insere em uma nova população usando um tipo de seleção natural juntamente com os operadores de cruzamento e mutação inspirados na genética. Cada cromossomo consiste de genes, que são parâmetros codificados nos cromossomos, ou seja, um elemento do vetor que representa o cromossomo, e cada gene representa uma instância de um alelo particular (valores que o gene pode assumir). O operador de seleção escolhe, dentre os cromossomos da população, aqueles que irão se

reproduzir. Em média, os cromossomos mais fortes (melhores adaptados) produzem mais descendentes do que aqueles mais fracos (menos adaptados). Os cromossomos selecionados podem sofrer modificações em suas características através dos operadores de cruzamento e mutação, gerando descendentes para a próxima geração (Mitchell, 1996).

Em se tratando de propósitos computacionais, um algoritmo genético simples possui uma estrutura conforme o Algoritmo 4.1, apresentado em (Michalewicz, 1996):



Algoritmo 4.1: Estrutura de um algoritmo genético.

Durante uma iteração t , o algoritmo genético mantém uma população de soluções candidatas (cromossomos), $P(t) = \{x_1^t, \dots, x_n^t\}$. Cada solução x_i^t é avaliada para medir sua aptidão (*fitness*), ou seja, a qualidade da solução do problema representada por este cromossomo. Então, uma nova população (iteração $t + 1$) é formada pela seleção dos indivíduos (cromossomos) mais aptos. Alguns membros desta nova população serão selecionados para fazer parte da chamada ‘população intermediária’, a qual sofrerá alterações devido à ação de operadores genéticos, enquanto outros permanecerão intactos.

4.1 Exploração e Prospecção

Qualquer método de busca deve usar as técnicas de exploração e prospecção para encontrar um ótimo global (Beasley et al. 1993). A exploração é o processo de visitar pontos inteiramente novos ou explorar regiões desconhecidas do espaço de busca. Já a prospecção, é o processo de explorar o espaço de busca utilizando informações de pontos anteriormente visitados, a fim de encontrar melhores pontos.

Os algoritmos genéticos combinam estas duas técnicas simultaneamente, através do uso de métodos de seleção e operadores genéticos. Os métodos de seleção realizam a prospecção, enquanto que os operadores genéticos de cruzamento e mutação fazem a exploração do espaço de busca.

Um fator que influencia a quantidade de exploração e prospecção no AG é a pressão de seleção, definida como a razão entre a aptidão máxima e a aptidão média. Quando a pressão de seleção é maior, temos mais prospecção, sendo que os melhores indivíduos têm valores de aptidão muito altos. Assim, estes indivíduos tendem a dominar as populações seguintes, uma vez que seus genes se propagarão através das gerações com alta probabilidade. Desta forma, o algoritmo genético converge rapidamente (provavelmente para um mínimo local) sem explorar pontos desconhecidos do espaço de busca. Por outro lado, quando a pressão de seleção é menor, o AG tem um comportamento similar a um método de busca randômica (muita exploração), pois os valores de aptidão dos indivíduos são praticamente iguais.

Em resumo, se a pressão de seleção é baixa, o AG converge lentamente, mas o espaço de busca é completamente explorado. Quando a pressão é alta, o AG converge rapidamente, mas não explora o espaço de busca de maneira abrangente. A análise destes fatores permite concluir que a convergência precisa do algoritmo genético depende do equilíbrio entre exploração e prospecção.

4.2 Métodos e Parâmetros do Algoritmo Genético Desenvolvido

Segundo Michalewicz (1996), para a implementação de um algoritmo genético deve-se definir alguns aspectos importantes, tais como uma representação genética para as soluções do problema, uma forma de criar uma população inicial das soluções, definição do tamanho desta população, uma função de avaliação que desempenha o papel do ambiente, métodos de seleção de indivíduos, operadores genéticos que alteram a composição dos filhos e um critério de parada. A influência de cada um destes aspectos no desempenho do algoritmo depende da classe de problemas que está sendo tratada.

4.2.1 Representação Genética

Existem alguns tipos de representação genética para as soluções de um problema, dentre as quais destacam-se as codificações binária e real. A representação binária (cadeias de zeros e uns) é a tradicionalmente usada, uma vez que é de fácil utilização e manipulação, além de simples de analisar teoricamente. Contudo, apresenta algumas desvantagens quando aplicada a problemas multidimensionais e a problemas numéricos de alta precisão (Michalewicz, 1996).

Na fase inicial do desenvolvimento deste trabalho, a codificação binária foi utilizada, mas apresentou problemas. Primeiramente, ocorreu perda de precisão em função de conversão numérica entre bases (real, decimal e binária). Em seguida, foi observado que para problemas com muitas variáveis, os cromossomos se tornavam muito grandes, o que tornava mais lento o processo de busca da solução. Por fim, a natureza da codificação binária torna mais difícil a proposição de diferentes operadores genéticos. Para fins de comparação, uma execução da RHG utilizando um algoritmo genético com codificação binária é apresentada no Apêndice B.

Assim sendo, optou-se por trocar a codificação binária pela real. Na codificação real, o cromossomo é um vetor de números de ponto flutuante, no qual cada componente do vetor

(gene) representa uma variável ou parâmetro do problema. A codificação real possui um valor semântico maior do que a codificação binária, sendo de mais fácil interpretação humana. A codificação real também é mais compatível com os métodos de otimização tradicionais, pois os mesmos utilizam valores de ponto flutuante em sua execução. Além disso, conforme mencionado em Lacerda (2003), a codificação real tem apresentado melhor desempenho em problemas de otimização com variáveis contínuas.

4.2.2 População Inicial

O passo seguinte é fazer a escolha de um tamanho de população, que produza respostas corretas e rápidas. A escolha de uma população inicial maior que a população a ser utilizada nas gerações subsequentes pode melhorar a representação do espaço de busca. Se uma população inicial pequena for gerada aleatoriamente, provavelmente algumas regiões do espaço de busca não serão representadas (Lacerda e Carvalho, 1999).

Geralmente, utiliza-se um tamanho de população proporcional ao tamanho do cromossomo, isto é, quanto maior for o cromossomo maior deverá ser o tamanho da população a fim de manter uma diversidade razoável. O tamanho da população utilizada nesta tese foi de 100 indivíduos, pois possibilitou uma melhor cobertura do espaço de busca e se mostrou eficiente nos experimentos realizados (Capítulo 5).

Um dos cromossomos pertencente à população inicial é constituído pelos valores de v previamente obtidos com a aplicação da rede neural, por meio dos passos (I) e (II) (descritos na Seção 3.1.3); 10% dos cromossomos foram gerados em torno deste v repassado pela rede neural, objetivando inserir uma tendência a regiões supostamente promissoras do espaço de busca. Os demais cromossomos foram gerados aleatoriamente.

4.2.3 Função de Avaliação

Uma função de avaliação deve desempenhar o papel do ambiente, ou seja, avaliar as soluções em termos de aptidão. Segundo Lacerda e Carvalho (1999), alguns cuidados devem ser tomados para que cromossomos idênticos não sejam avaliados mais de uma vez, tais como: evitar gerar cromossomos idênticos na população inicial, verificar se o filho é igual a um dos pais e, antes de avaliar um filho, verificar se um cromossomo idêntico já existe na população.

A função de avaliação do algoritmo genético para problemas de otimização não-linear restrita é a própria função objetivo do problema a ser otimizado. Como se tratam de problemas de minimização, será mais adaptado o indivíduo que possuir menor valor de aptidão, ou seja, o indivíduo que proporcionar um menor valor para a função objetivo.

4.2.4 Métodos de Seleção

Dada uma população em que a cada indivíduo foi atribuído um valor de aptidão, existem vários métodos para selecionar os indivíduos sobre os quais serão aplicados os operadores genéticos de cruzamento e mutação. Estes indivíduos selecionados formarão uma população, que é conhecida como população intermediária. A maioria dos métodos de seleção é projetada para escolher preferencialmente indivíduos com maiores ou menores valores de aptidão, embora não exclusivamente, a fim de manter a diversidade da população.

O método de seleção utilizado neste trabalho para selecionar a população intermediária foi o método do Torneio, no qual n indivíduos da população são escolhidos aleatoriamente com a mesma probabilidade. O cromossomo com melhor aptidão dentre estes é selecionado para a população intermediária. O processo se repete até que a população intermediária seja preenchida. Um exemplo da implementação em pseudocódigo deste método é descrito pelo Algoritmo 4.2:

<p>Início <i>repita r vezes para selecionar r indivíduos</i> <i>Escolhe-se n indivíduos da população aleatoriamente</i> <i>Indivíduo com melhor aptidão dentre estes n é selecionado</i> <i>fim repita</i> Fim</p>
--

Algoritmo 4.2: Método de seleção por Torneio

O valor escolhido para n foi 2, comparando os valores de aptidão de dois indivíduos por vez. Como os problemas tratados nesta tese são problemas de minimização, o indivíduo mais apto é aquele que possui o menor valor de aptidão.

4.2.5 Operadores Genéticos

Os operadores genéticos de cruzamento e mutação provocam alterações em uma população. O operador de cruzamento é aplicado aos cromossomos de acordo com uma probabilidade conhecida como taxa de cruzamento. Quanto maior for esta taxa, mais rapidamente novos indivíduos serão introduzidos na população. Mas se esta for muito alta, a maior parte da população será substituída e poderá ocorrer a perda de indivíduos com alta aptidão. Com um valor baixo, o algoritmo pode se tornar muito lento. A mutação é necessária para a introdução e manutenção da diversidade genética da população, alterando arbitrariamente um ou mais indivíduos, fornecendo assim, meios para a introdução de novos indivíduos na população. O operador de mutação é aplicado aos indivíduos com uma probabilidade dada pela taxa de mutação. Uma baixa taxa de mutação pode assegurar a diversidade na população, mas o contrário pode destruir toda a informação contida no indivíduo que foi adquirida durante as gerações passadas.

Assim, as taxas de cruzamento e mutação devem ser definidas empiricamente para cada domínio de problema, levando em conta as considerações mencionadas. A maior parte da literatura recomenda uma taxa de cruzamento entre 75% e 95% (Mitchell, 1996). A taxa de cruzamento utilizada nesta tese foi de 75%, valor usual que se mostrou adequado para os

experimentos. A literatura ainda recomenda uma taxa de mutação entre 0,5% e 1% (Mitchell, 1996). O valor da taxa de mutação utilizada foi de 1%, estando dentro da faixa recomendada.

A escolha dos operadores genéticos está intimamente ligada à codificação adotada para a representação genética (Delgado, 2002), ou seja, existe uma variação no comportamento dos operadores, o qual está relacionado à codificação empregada nos cromossomos. Em outras palavras, há operadores genéticos que foram concebidos para o uso com codificação binária e operadores genéticos para o uso com codificação real ou inteira. Com base em Michalewicz (1996), serão definidos apenas os operadores genéticos para codificação real utilizados no desenvolvimento deste trabalho.

Operadores Genéticos para Codificação Real

Cruzamento BLX- α

O cruzamento BLX- α consiste em gerar um novo cromossomo a partir da seguinte expressão:

$$c = p_1 + r(p_2 - p_1) \quad (4.1)$$

sendo que c é o cromossomo filho gerado, p_1 e p_2 são os cromossomos pais e $r \in U(-\alpha, 1 + \alpha)$. O termo α é um pequeno valor que estende os limites para a definição de c . Se $\alpha = 0$, os cromossomos filhos estarão dentro do segmento de linha L (mostrado na Figura 4.1) que une p_1 e p_2 . Se $\alpha > 0$, o segmento de linha L é estendido. Usualmente, define-se $\alpha = 0.5$ (estende o segmento L em $0.5L$ para cada extremo), possibilitando que os filhos estejam dentro ou fora do segmento de linha L com a mesma probabilidade. Caso o cromossomo seja formado por múltiplos genes, a Equação (4.1) é aplicada a cada par de genes de p_1 e p_2 .

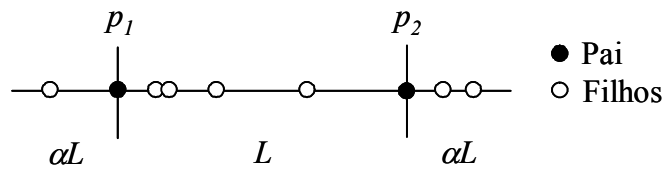


Figura 4.1: Cruzamento BLX- α

Mutação Uniforme

O operador de mutação uniforme seleciona aleatoriamente um gene do cromossomo e substitui seu valor por um número aleatório gerado dentro do domínio definido para a variável em questão. O comportamento deste operador é ilustrado na Figura 4.2.

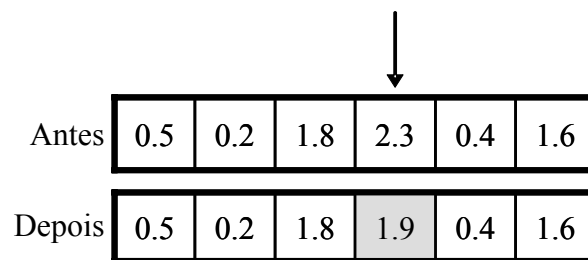


Figura 4.2. Mutação uniforme.

4.2.6 Critério de Parada

Foi utilizado como critério de parada para o algoritmo genético o desvio padrão entre os valores de aptidão dos indivíduos da população. Quando o desvio padrão atinge uma precisão requerida, o algoritmo genético é finalizado.

4.3 Considerações Finais

Este capítulo apresentou uma visão geral sobre os componentes de um algoritmo genético (representação genética, população inicial, função de avaliação, operadores genéticos

e critério de parada) e descreveu os critérios utilizados para a aplicação desta metodologia na solução de problemas de otimização não-linear tratados nesta tese.

Após a minimização de E^{conf} efetuada pelos passos (I) e (II) da Rede de Hopfield Genética proposta, os valores de v são repassados ao algoritmo genético. Estes valores são utilizados como uma possível solução para o problema de otimização, sendo inseridos como um cromossomo na população inicial. O algoritmo genético é colocado em execução, e ao finalizar, devolve novos valores de v , os quais representam um ponto de mínimo para a função objetivo do problema em questão. O processo continua repetidamente, até que a rede neural e o algoritmo genético “concordem” com uma solução, considerando o nível de precisão exigido.

CAPÍTULO 5

Experimentos e Resultados

Como visto anteriormente, para sistemas não-lineares, tem-se que em uma pequena vizinhança em torno dos pontos de equilíbrio, o sistema não-linear se comporta como linear. Neste caso, verifica-se que a matriz T^{conf} se torna constante em torno do ponto de equilíbrio $x^e = 0$. Assim, a abordagem neuro-genética, composta pela rede neural de Hopfield modificada (cujo subespaço-válido é calculado por (3.38) e (3.39)) e pelo algoritmo genético (o qual mapeia a função objetivo), quando aplicada a um problema de otimização não-linear da forma dado por (3.29)-(3.32), sempre converge para uma solução válida.

A seguir, alguns experimentos e seus respectivos resultados são apresentados a fim de demonstrar a efetividade da Rede de Hopfield Genética quando utilizada na obtenção de soluções de problemas de otimização não-linear restrita.

Todos os experimentos são baseados em problemas apresentados nos trabalhos descritos no Capítulo 2, e foram realizados utilizando um processador Intel Core 2 Duo de 1.8GHz e memória RAM de 2 Gigabytes. Esta informação é importante para que se possa analisar o tempo de convergência, em nível de CPU. No entanto, comparações são difíceis de serem estabelecidas, pois a maioria dos trabalhos que tratam de problemas de otimização não-linear restrita não apresentam esta informação.

5.1 Experimentos Baseados em Redes Neurais Artificiais

Serão apresentados nesta seção, os resultados obtidos pela Rede de Hopfield Genética em alguns experimentos e sua comparação com outras abordagens que utilizam redes neurais artificiais (citadas no Capítulo 2) em otimização restrita.

5.1.1 Experimento 1

Seja o problema de otimização convexa restrita com duas restrições de desigualdade lineares:

$$\text{Min } f(v) = 0,4v_2 + v_1^2 + v_2^2 - v_1 \cdot v_2 + \frac{1}{30} v_1^3$$

$$\text{Sujeito a: } v_1 + 0,5v_2 \geq 0,4$$

$$0,5v_1 + v_2 \geq 0,5$$

$$v_1, v_2 \geq 0$$

Após a convergência da Rede de Hopfield Genética, o vetor obtido é dado por $v = [0,3378 \ 0,3300]^T$, com $f(v) = 0,2448$. Estes resultados são bem próximos aos valores da solução exata fornecida por $v^* = [0,3395 \ 0,3302]^T$, com função objetivo $f(v^*) = 0,2455$.

A Figura 5.1 e a Figura 5.2 apresentam a evolução do vetor de saída da RHG e o comportamento da função objetivo do problema, respectivamente.

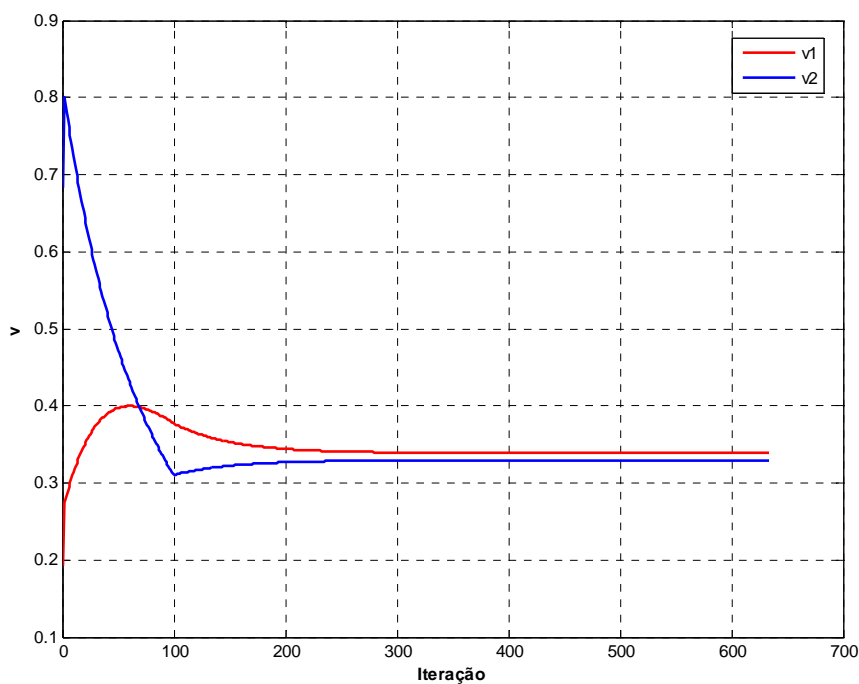


Figura 5.1: Evolução do vetor de saída da RHG.

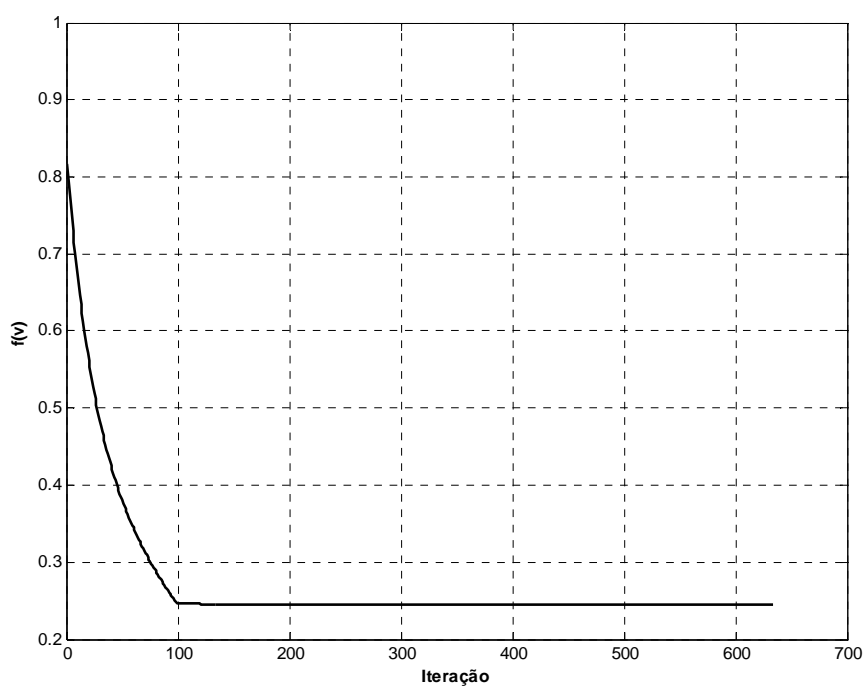


Figura 5.2: Comportamento da função objetivo.

A Rede de Hopfield Genética foi avaliada inicializando de diferentes pontos. A Figura 5.3 ilustra o comportamento do vetor v partindo de dois diferentes pontos iniciais e convergindo para a solução do problema de otimização restrita.

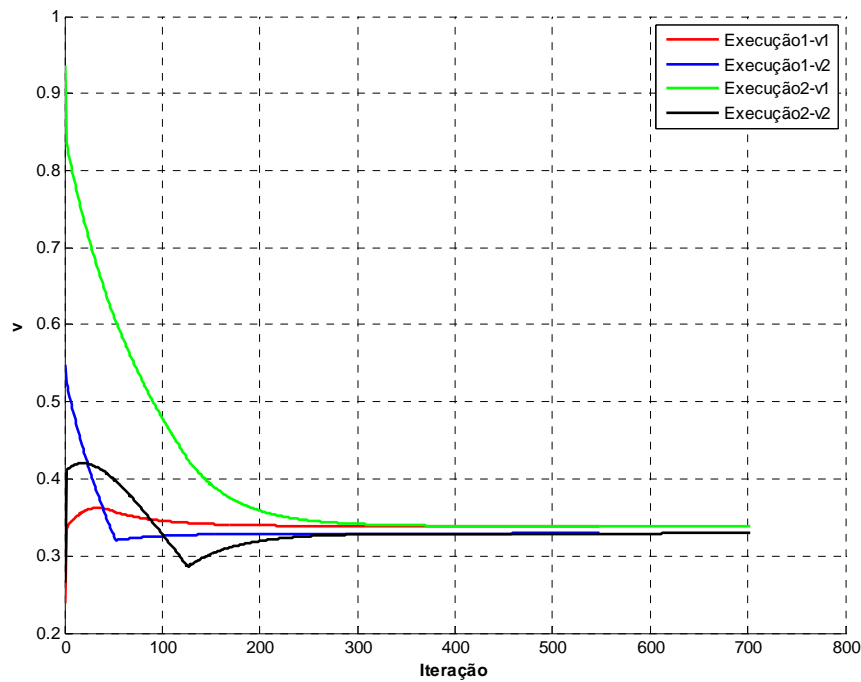


Figura 5.3: Vetor v em duas execuções diferentes.

A Execução 1 partiu do ponto $v = [0,2389 \ 0,5481]^T$ e atingiu o ponto $v = [0,3378 \ 0,3300]^T$, com $f(v) = 0,2448$, após 701 iterações e com um tempo de CPU de 5,15 segundos. A Execução 2 iniciou no ponto $v = [0,9363 \ 0,2651]^T$ e atingiu o ponto $v = [0,3353 \ 0,3222]^T$, com $f(v) = 0,2383$, após 547 iterações e com um tempo de CPU de 3,62 segundos. Assim, apesar da Execução 2 ter obtido um valor um pouco pior para $f(v)$, o número de iterações e o tempo gasto na execução foram consideravelmente menores.

A Figura 5.4 mostra o comportamento da função objetivo para ambas as execuções.

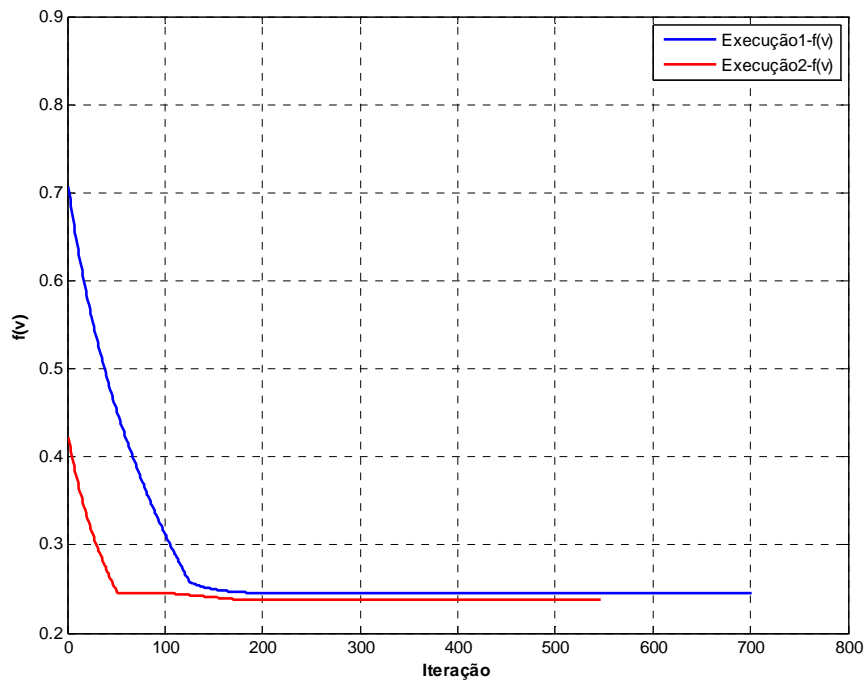


Figura 5.4: Valor da função objetivos para duas execuções diferentes.

Este problema também foi solucionado pelos autores Kennedy e Chua (1988) e Silva (1997). Kennedy e Chua, (1988) propuseram uma rede neural que utiliza parâmetros de ponderação no mapeamento de problemas de otimização restrita, os quais fazem com que a rede produza apenas soluções aproximadas, com risco de elevado custo computacional caso seus valores não sejam devidamente especificados. O resultado obtido por estes autores foi $v = [0,3406 \ 0,3385]^T$, com $f(v) = 0,2520$. Em Silva (1997) é apresentada uma rede recorrente para resolver problemas de otimização restrita, sendo que os valores encontrados para este problema, após 364 iterações, foram $v = [0,3398 \ 0,3301]^T$, com $f(v) = 0,2456$. Entretanto, esta abordagem possui a desvantagem de utilizar um parâmetro de inicialização, diferente para cada problema, o qual é obtido experimentalmente.

5.1.2 Experimento 2

Seja o problema de otimização não-convexa não-linear, definido em Silva (1997), composto por uma restrição de igualdade e uma de desigualdade:

$$\begin{aligned} \text{Min } f(v) &= v_1^3 + 2v_2^2 \cdot v_3 + 2v_3 \\ \text{Sujeito a: } & v_1^2 + v_2 + v_3^2 = 4 \\ & v_1^2 - v_2 + 2v_3 \leq 2 \\ & v_1, v_2, v_3 \geq 0 \end{aligned}$$

O vetor solução obtido, após a convergência da Rede de Hopfield Genética, é dado por: $v = [0,0000 \ 3,9779 \ 0,0000]^T$, com $f(v)$ bem próxima de 0. A solução ótima para o problema é fornecida por $v^* = [0,0000 \ 4,0000 \ 0,0000]^T$, com $f(v^*) = 0$. A Figura 5.5 mostra a evolução dos valores de v_1 , v_2 e v_3 em relação ao número de iterações.

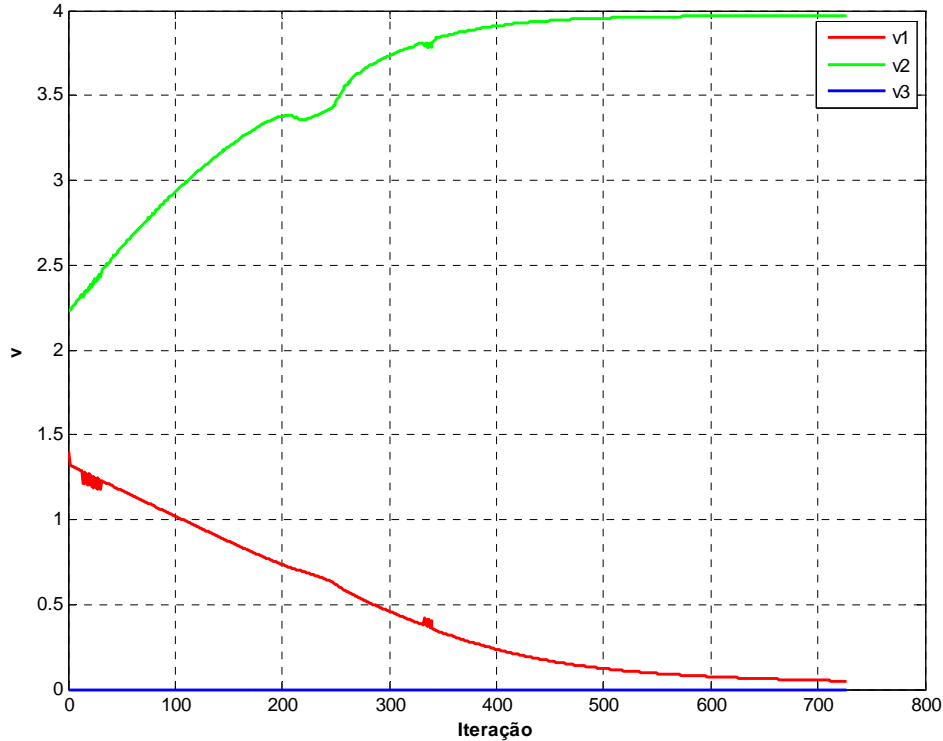


Figura 5.5: Evolução do vetor de saída da RHG.

Foram realizadas 10 execuções com a RHG, partindo de diferentes valores iniciais do vetor v e incluindo pontos fora do domínio definido para o problema. Todas as trajetórias seguiram rumo ao mesmo ponto de equilíbrio. A Tabela 5.2 mostra o resultado destas 10 execuções, apresentando os valores obtidos para o vetor v e para $f(v)$, o número de iterações e o tempo de CPU (em segundos) gasto em cada execução. Nas execuções 3, 5 e 9, os pontos iniciais escolhidos foram constituídos por valores menores que zero, extrapolando o domínio definido para as variáveis do problema ($v_1, v_2, v_3 \geq 0$). Isto explica o número elevado de iterações para encontrar a solução.

Tabela 5.1: Resultados utilizando a RHG.

	v_1	v_2	v_3	$f(v)$	Número de Iterações	Tempo de CPU (s)
Execução 1	0,0000	4,0006	0,0000	0	568	14,82
Execução 2	0,0000	3,9991	0,0000	0	351	7,62
Execução 3	0,0000	3,9994	0,0000	0	1818	36,01
Execução 4	0,0000	3,9994	0,0000	0	727	15,64
Execução 5	0,0000	3,9994	0,0000	0	1471	29,62
Execução 6	0,0000	3,9992	0,0000	0	289	6,09
Execução 7	0,0000	4,0006	0,0000	0	661	13,71
Execução 8	0,0000	3,9991	0,0000	0	90	2,10
Execução 9	0,0000	4,0005	0,0000	0	1142	23,03
Execução 10	0,0000	3,9992	0,0000	0	237	5,25
Média				0	735,4	15,38

Considerando que todas as execuções alcançaram resultados bem próximos ao ótimo, a decisão sobre qual execução é a melhor depende da necessidade da aplicação. Se o aspecto mais importante for a precisão, a Execução 9 é a melhor solução. Já para os casos em que se buscam respostas rápidas, a Execução 8 seria a melhor escolha.

A Figura 5.6 ilustra o comportamento da função objetivo do problema em relação ao número de iterações. Os valores iniciais atribuídos ao vetor v foram gerados aleatoriamente dentro do domínio definido para as variáveis.

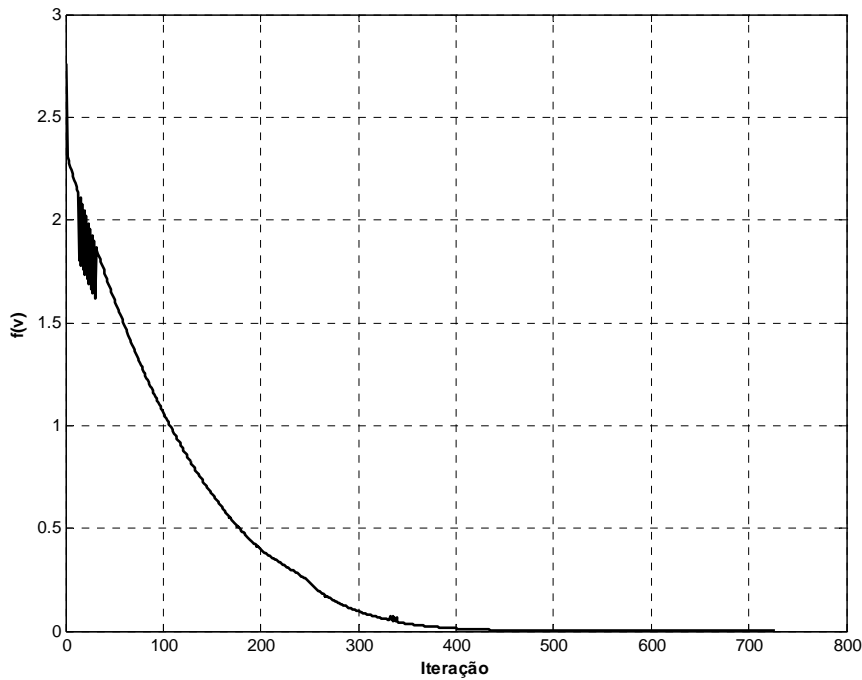


Figura 5.6: Comportamento da função objetivo.

Em Silva (1997) foi obtido o resultado $v = [0,000 \ 3,997 \ 0,000]^T$, cujo valor da função objetivo vale $f(v) = 0,0001$. Esta resposta foi encontrada somente após 953 iterações.

5.1.3 Experimento 3

Seja o problema de otimização não-linear restrita composto por uma função objetivo não-convexa, restrições de desigualdade e variáveis canalizadas, proposto em Silva (1997):

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

$$\text{Sujeito a: } v_1^2 + e^{v_2} + 6v_3 \leq 15$$

$$v_1^4 - v_2 + 5v_3 \leq 25$$

$$v_1^3 + v_2^2 - v_3 \leq 10$$

$$0 \leq v_1 \leq 4$$

$$0 \leq v_2 \leq 2$$

$$v_3 \geq 0$$

Este problema tem uma única solução ótima dada por $v^* = [0,0000 \ 1,5000 \ 0,0000]^T$, com função objetivo $f(v^*) = -3,5000$. A Figura 5.7 mostra a trajetória da RHG, iniciando no ponto $v_0 = [0,3624 \ 0,2391 \ 0,6121]^T$ e convergindo para o ponto $v = [0,0000 \ 1,5022 \ 0,0005]^T$, cujo valor da função objetivo vale $f(v) = -3,4991$.

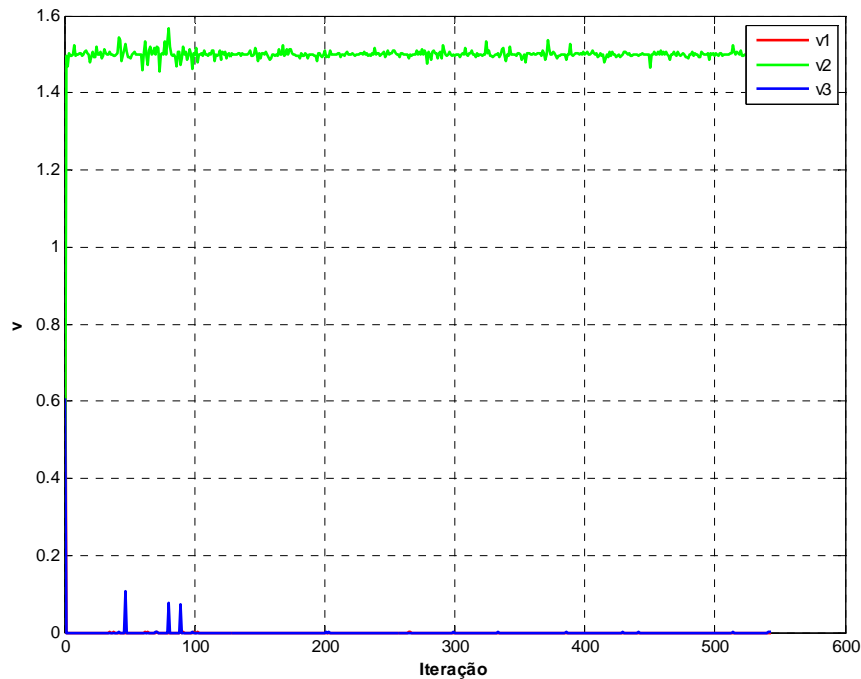


Figura 5.7: Evolução do vetor de saída da RHG.

As variáveis canalizadas representadas pelas três últimas restrições são mapeadas diretamente através da função de ativação rampa-simétrica, definida em (3.11). A Rede de Hopfield Genética foi também avaliada para diferentes valores referentes às condições iniciais do vetor v , incluindo valores fora do domínio definido para cada variável do problema. Todas as trajetórias seguiram rumo ao mesmo ponto de equilíbrio. A Tabela 5.2 mostra o resultado de 10 execuções da RHG, apresentando os valores obtidos para o vetor v e $f(v)$, o número de iterações e o tempo de CPU (em segundos) usado em cada execução.

Tabela 5.2: Resultados utilizando a RHG.

	v_1	v_2	v_3	$f(v)$	Número de Iterações	Tempo de CPU (s)
Execução 1	0,0000	1,5028	0,0009	-3,4982	542	26,14
Execução 2	0,0000	1,4990	0,0007	-3,4985	578	29,58
Execução 3	0,0000	1,4981	0,0017	-3,4966	517	24,82
Execução 4	0,0000	1,5022	0,0005	-3,4991	547	29,54
Execução 5	0,0000	1,4999	0,0067	-3,4866	425	20,06
Execução 6	0,0000	1,4970	0,0008	-3,4985	591	21,12
Execução 7	0,0000	1,5005	0,0007	-3,4987	628	22,29
Execução 8	0,0000	1,4972	0,0006	-3,4987	667	23,43
Execução 9	0,0000	1,4959	0,0005	-3,4990	777	26,62
Execução 10	0,0000	1,5028	0,0009	-3,4982	542	27,20
Média				-3,4972	581,4	25,08

A Figura 5.8 ilustra o comportamento da função objetivo do problema em função do número de iterações.

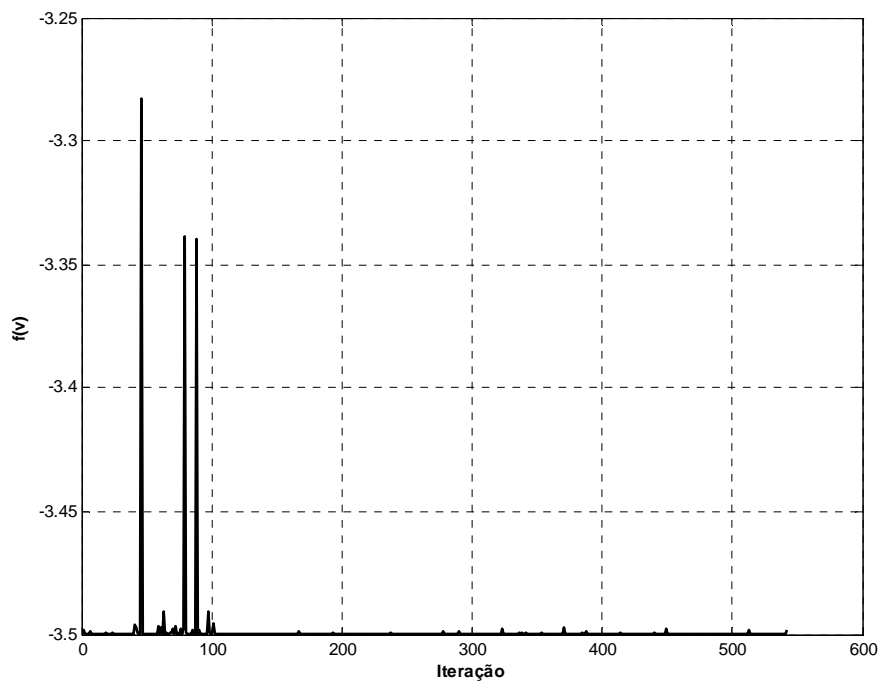


Figura 5.8: Comportamento da função objetivo.

Em Silva (1997), os valores encontrados para v e $f(v)$ foram $v = [0,0022 \ 1,5008 \ 0,0000]^T$ e $f(v) = -3,555$, com 1621 iterações, o que demonstra a eficácia do método e mostra que a convergência em direção aos pontos de equilíbrio é lenta. Além disso, a abordagem

proposta utiliza um parâmetro de inicialização, o qual é definido experimentalmente para cada problema.

5.1.4 Experimento 4

Considere o problema de otimização convexa apresentado em Zheng et al. (2002):

$$\begin{aligned} \text{Min } f(v) &= -v_1 \\ \text{Sujeito a: } & v_2 - v_1^3 - v_3^2 = 0 \\ & v_1^2 - v_2 - v_4^2 = 0 \\ & v_2 - v_1^3 \geq 0 \\ & v_1^2 - v_2 \geq 0 \end{aligned}$$

Este problema é composto por duas restrições de igualdade e duas de desigualdade. A solução ótima descrita na literatura para este problema é $v^* = [1,0000 \ 1,0000 \ 0,0000 \ 0,0000]^T$, com função objetivo $f(v^*) = -1,0000$. A RHG alcançou exatamente este valor ótimo, após 463 iterações, com tempo de CPU de 3,72 segundos. A Figura 5.9 e a Figura 5.10 apresentam a evolução do vetor v e de $f(v)$, respectivamente.

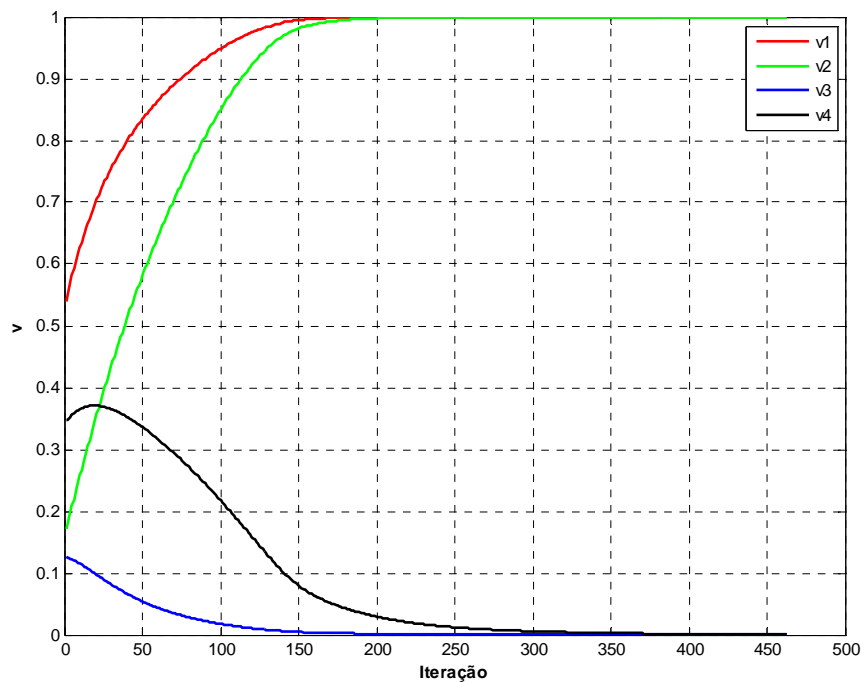


Figura 5.9: Evolução do vetor de saída da RHG.

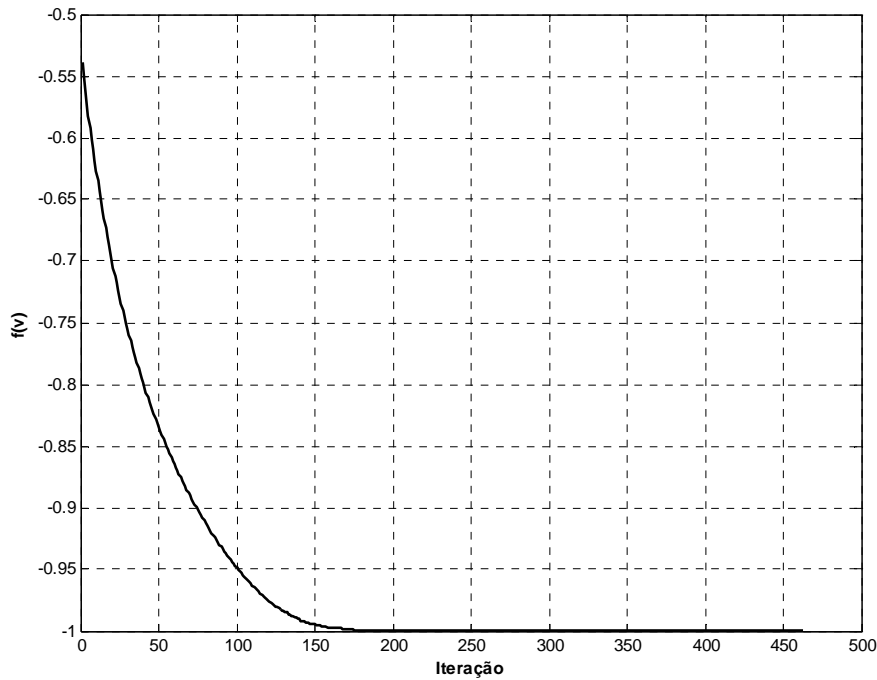


Figura 5.10: Comportamento da função objetivo.

A abordagem apresentada por Zheng et al. (2002) obteve uma solução aproximada, com $v = [1,0020 \ 1,0006 \ 0,0000 \ 0,0000]^T$ e $f(v) = -1,0020$, realizando para isso 10566 iterações. O tempo de processamento é bastante elevado, em função da necessidade de se obter os valores de três vetores que representam parâmetros de ponderação.

5.1.5 Experimento 5

Lua e Itob (2003) utilizaram sua abordagem na resolução do problema convexo apresentado a seguir, cuja solução ótima vale $v^* = [-1,0144 \ -0,0440]^T$ e $f(v^*) = 11,7127$.

$$\text{Min } f(v) = v_1^2 - v_2^2 - 2v_1v_2 + 8v_1 - 3v_2 + e^{-v_1} + 16$$

$$\text{Sujeito a: } -6v_1^2 + 9v_1v_2 - 4v_2^2 + 5v_2 + 6 \geq 0$$

$$2v_1 - 2v_2 + 3 \geq 0$$

$$-2 \leq v_1 \leq 2$$

$$-2 \leq v_2 \leq 2$$

Como resultado, foram encontrados os valores $v = [-1,0707 \ -0,2426]^T$ e $f(v) = 11,6477$. Os autores destacam que apenas soluções aproximadas são obtidas, uma vez que, um problema não-linear é decomposto em problemas lineares, utilizando aproximação funcional.

Também foram encontrados apenas resultados aproximados com a aplicação da Rede de Hopfield Genética, os quais valem $v = [-1,0396 \ -0,0362]^T$ e $f(v) = 11,6246$. A Figura 5.11 apresenta a evolução do vetor v de saída da RHG.

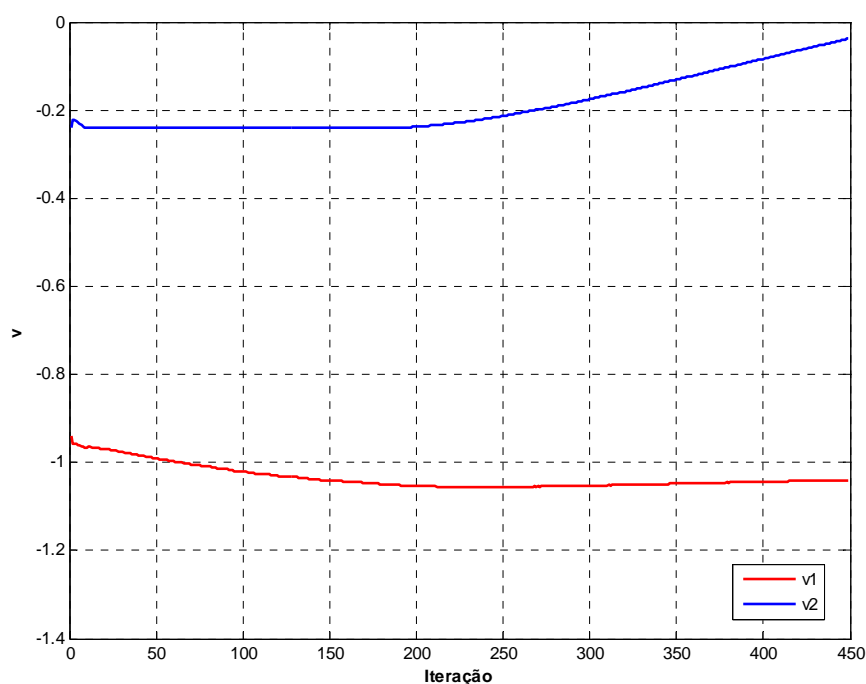


Figura 5.11: Evolução do vetor de saída da RHG.

O número de iterações realizadas para a obtenção de tais resultados foi de 449, com tempo de CPU de 8,22 segundos. A Figura 5.12 mostra o comportamento da função objetivo durante o processo de convergência.

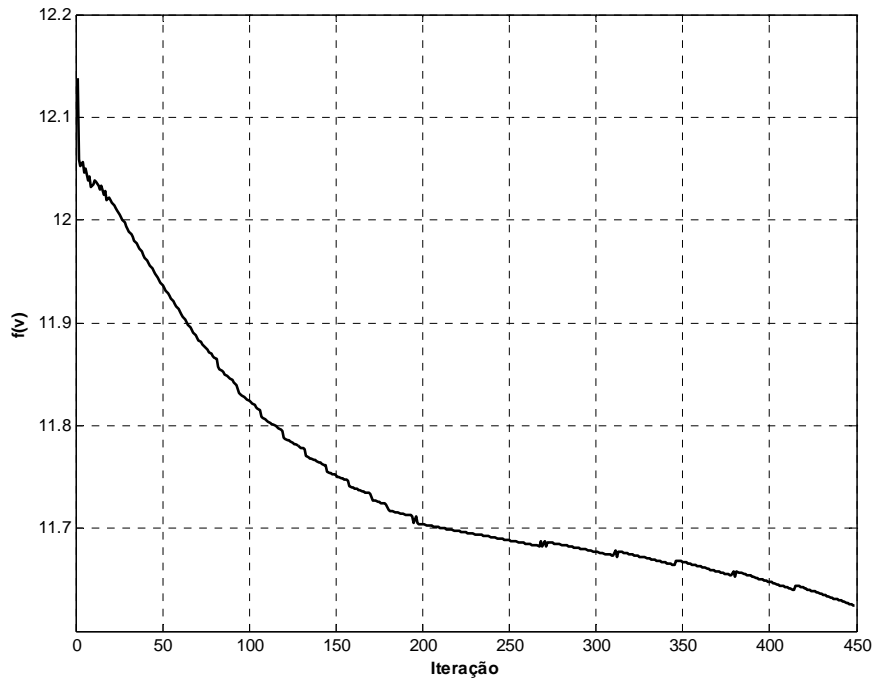


Figura 5.12: Comportamento da função objetivo.

Apesar de o resultado ter sido um pouco pior que o encontrado por Lua e Itob (2003), vale destacar a generalidade da RHG. Além disso, os autores não mencionam número de iterações ou tempo de execução, impossibilitando qualquer outra comparação.

5.1.6 Experimento 6

Seja o problema de otimização convexa não-linear apresentado em Xia e Wang (2003):

$$\text{Min } f(v) = 0,25v_1^4 + 0,5v_1^2 + 0,25v_2^4 + 0,5v_2^2 - 0,9 v_1v_2$$

$$\text{Sujeito a: } \begin{aligned} v_1 + v_2 &\leq 2 \\ -v_1 + v_2 &\leq 2 \\ v_1 - 3v_2 &\leq -2 \\ v_1, v_2 &\geq 0 \end{aligned}$$

Este problema possui solução ótima $v^* = [0,0000 \ 1,9600]^T$ e $f(v^*) = 6,0115$. Xia e Wang (2003) obtiveram o resultado ótimo em seus experimentos para este problema. No entanto, os mesmos destacam que a abordagem desenvolvida pode convergir para ótimos locais. Já os resultados alcançados pela Rede de Hopfield Genética ficaram bem próximos do ótimo, sendo $v = [0,0000 \ 2,0000]^T$ e $f(v) = 6,0000$, com um número médio de iterações igual a 26 e tempo de CPU de 0,17 segundo, não convergindo para ótimos locais. A Figura 5.13 e a Figura 5.14 mostram, respectivamente, o comportamento do vetor v e de $f(v)$.

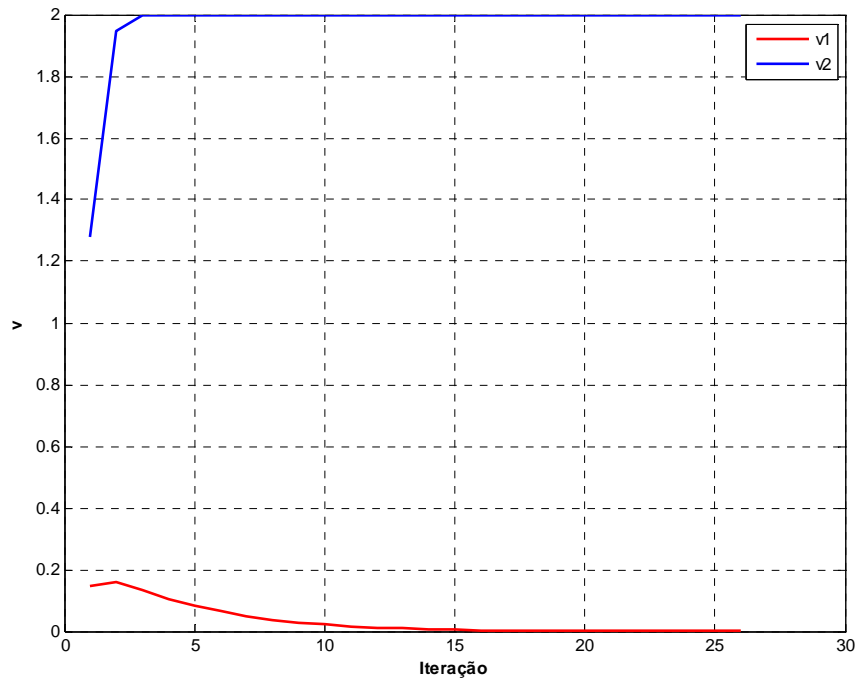


Figura 5.13: Evolução do vetor de saída da RHG.

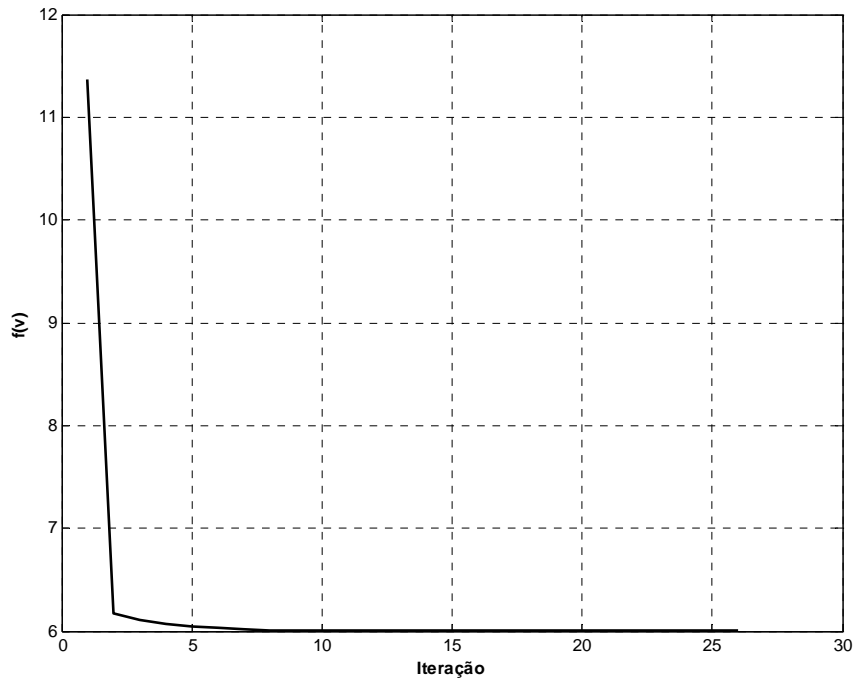


Figura 5.14: Comportamento da função objetivo.

5.1.7 Experimento 7

Considere o problema de otimização a seguir, composto por dez variáveis:

$$\text{Min } f(v) = v_1^2 + v_2^2 + (v_3 - 10)^2 + 4(v_4 - 5)^2 + (v_5 - 3)^2 + 2(v_6 - 1)^2 + 5v_7^2 + 7(v_8)^2 + 2(v_9)^2 + (v_{10} - 7)^2 + v_1v_2 - 14v_1 - 16v_2 + 45$$

$$\text{Sujeito a: } 3(v_1 - 2)^2 + 4(v_2 - 3)^2 + 2v_3^2 - 7v_4 - 120 \leq 0$$

$$5v_1^2 + (v_3 - 6)^2 + 8v_2 - 2v_4 - 40 \leq 0$$

$$(v_1 - 8)^2/2 + 2(v_2 - 4)^2 + 3v_5^2 - v_6 - 30 \leq 0$$

$$v_1^2 + 2(v_2 - 2)^2 - 2v_1v_2 + 14v_5 - 6v_6 \leq 0$$

$$4v_1 + 5v_2 - 3v_7 + 9v_8 - 105 \leq 0$$

$$10v_1 - 8v_2 - 17v_7 + 2v_8 \leq 0$$

$$12(v_9 - 8)^2 - 3v_1 + 6v_2 - 7v_{10} \leq 0$$

$$-8v_1 + 2v_2 + 5v_9 - 2v_{10} - 12 \leq 0$$

$$v_i \geq 0, (1 \leq i \leq 10)$$

Este problema possui muitas variáveis, o que o torna mais difícil de resolver que os demais problemas apresentados anteriormente. A solução ótima descrita na literatura é $v^* = [2,172 \ 2,364 \ 8,774 \ 5,096 \ 0,091 \ 1,431 \ 1,321 \ 9,829 \ 8,280 \ 8,376]^T$ e $f(v^*) = 826,5825$. A solução obtida pela RHG foi $v = [2,4993 \ 2,4992 \ 8,9970 \ 5,0036 \ 0,9999 \ 1,4000 \ 1,3005 \ 9,8000 \ 8,2000 \ 7,0203]^T$ e $f(v) = 809,2876$. A Figura 5.15 apresenta o comportamento de v durante as iterações.

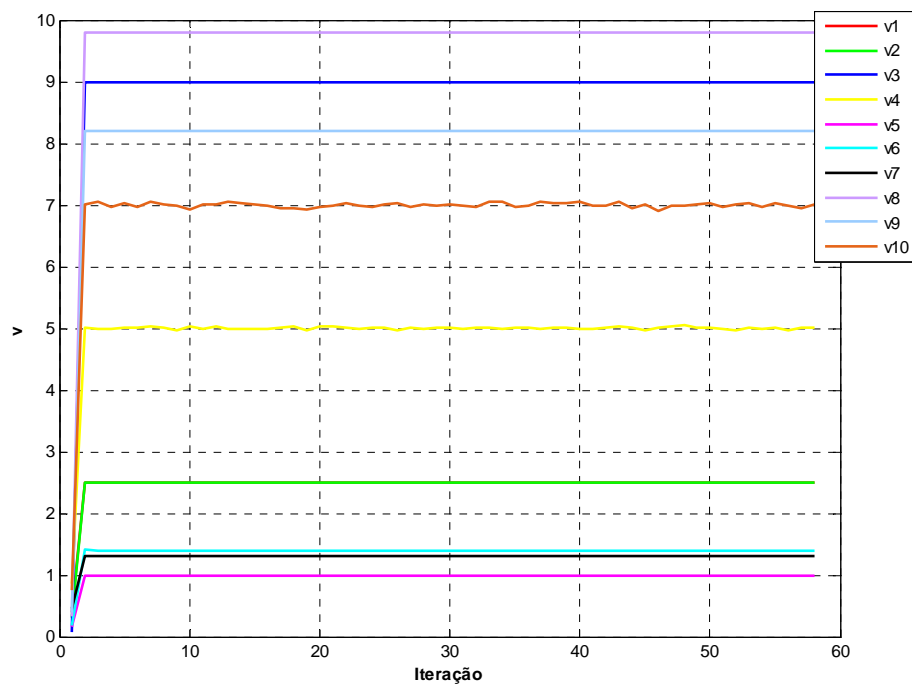


Figura 5.15: Evolução do vetor de saída da RHG.

Este resultado foi obtido após 58 iterações e com tempo de CPU igual a 4,36 segundos. Não houve melhora significativa no valor de $f(v)$ após 58 iterações. A Figura 5.16 mostra a evolução de $f(v)$.

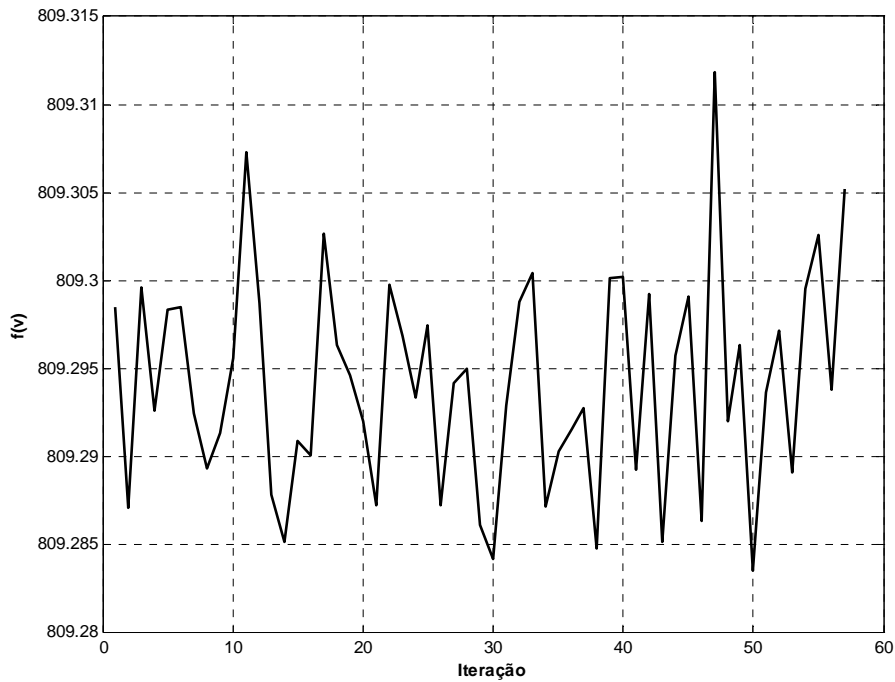


Figura 5.16: Comportamento da função objetivo.

Este problema também foi resolvido por Kennedy e Chua (1988) e por Xia e Wang (2004). Kennedy e Chua (1988) encontraram o resultado $v = [2,057 \ 2,3602 \ 8,905 \ 6,025 \ 1,236 \ 2,013 \ 1,258 \ 9,859 \ 10,875 \ 15,322]^T$ e $f(v) = 997,76$, o qual está bastante distante do valor ótimo. Os autores também mencionam um tempo de execução de 8 segundos, representando o dobro do tempo de execução da RHG.

Em seu trabalho, Xia e Wang (2004) obtiveram o resultado ótimo, com um tempo de execução de aproximadamente 4 segundos. Com base nestes resultados é possível observar que o modelo proposto por estes autores é eficiente em tratar problemas com muitas variáveis. No entanto, Xia e Wang (2004) destacam as elevadas complexidades de mapeamento e implementação. Além disso, o modelo proposto resolve apenas problemas de otimização convexa.

5.1.8 Experimento 8

Considere o problema de otimização apresentado em Ai et al. (2006):

$$\text{Min } f(v) = v_1^2 + 2v_2^2 - 2v_1v_2 - 2v_1 - 6v_2$$

$$\text{Sujeito a: } v_1 + v_2 \leq 2$$

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

Este problema apresenta solução ótima igual a $v^* = [0,8 \ 1,2]^T$ e $f(v^*) = -7,2$.

A Rede de Hopfield Genética encontrou a solução ótima, com tempo de CPU de 0,54 segundo e realizando 79 iterações. A Figura 5.17 e a Figura 5.18 apresentam os comportamentos de v e $f(v)$ no decorrer das iterações.

Ai et al. (2006) também encontrou a solução considerada ótima, com um tempo de execução de aproximadamente 0,7 segundos. Apesar dos bons resultados encontrados nos exemplos utilizados, esta abordagem é bastante restrita, resolvendo apenas problemas de otimização quadrática.

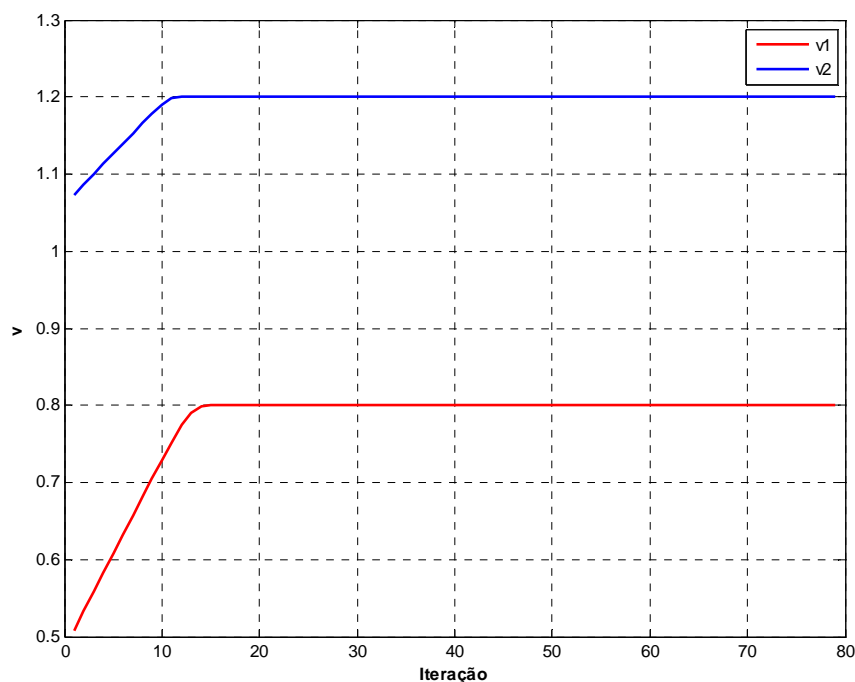


Figura 5.17: Evolução do vetor de saída da RHG.

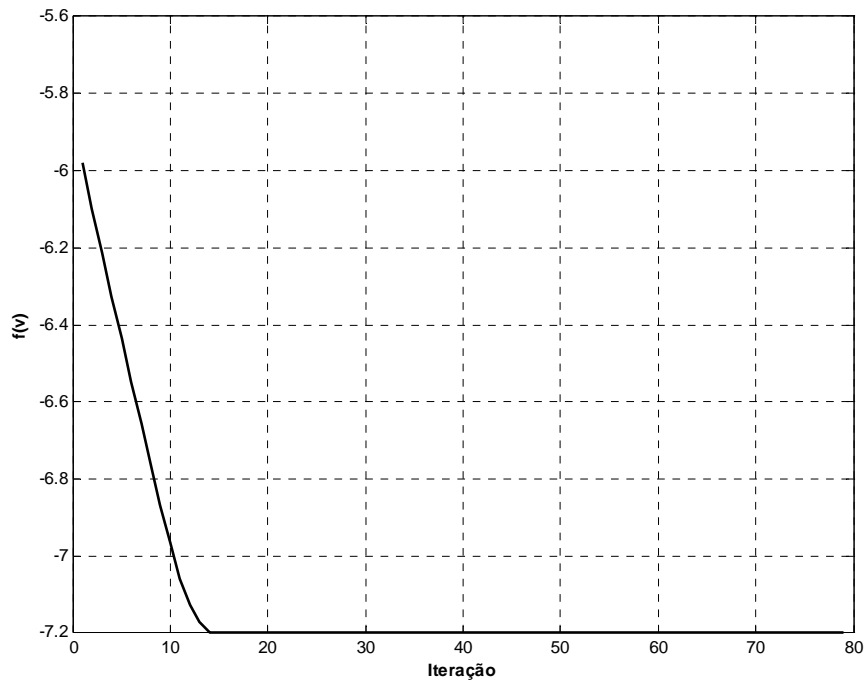


Figura 5.18: Comportamento da função objetivo.

5.1.9 Experimento 9

Considere o problema de otimização não-convexa apresentado em Hu e Wang (2006), cuja solução ótima é dada por $v^* = [-0,5143 \ -0,0572]^T$ e $f(v^*) = -0,3922$:

$$\text{Min } f(v) = v_1 \cdot \exp(-v_1^2 - 2v_2^2)$$

$$\text{Sujeito a: } -4v_1 + v_2 \leq 2$$

$$v \in V = \{v \in \mathbb{R}^2 \mid -2 \leq v_i \leq 2, i = 1,2\}$$

Em seu trabalho, Hu e Wang (2006) não apresentam os resultados obtidos por sua metodologia, apenas mencionam que nenhuma das trajetórias de execução alcançou exatamente v^* e que, caso o problema de otimização não obedeça a alguns critérios, a metodologia pode falhar.

A RHG alcançou resultados próximos ao ponto ótimo, com $v = [-0,6768 \ -0,0345]^T$ e $f(v) = -0,4271$. A Figura 5.19 mostra a evolução do vetor de saída da RHG.

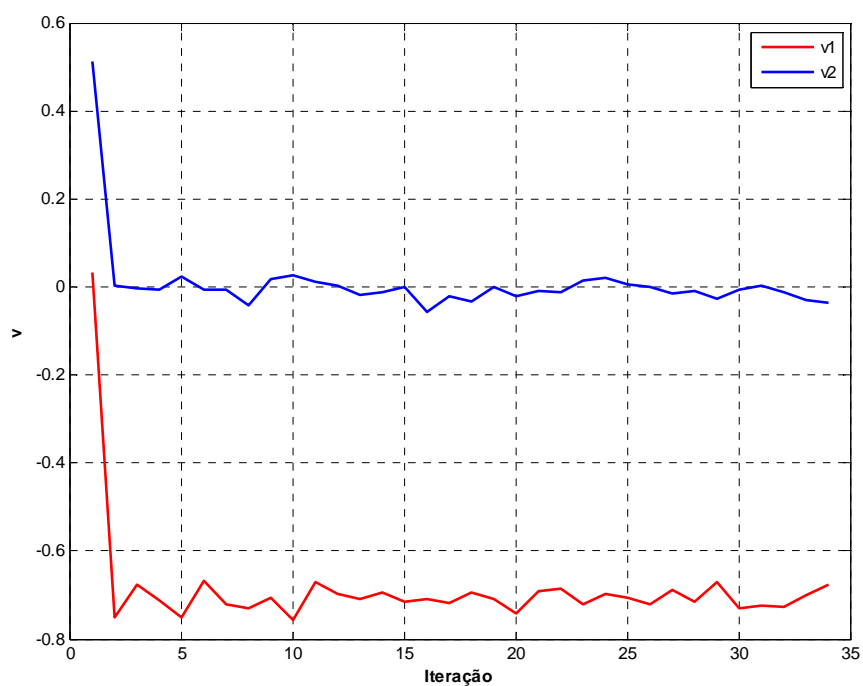


Figura 5.19: Evolução do vetor de saída da RHG.

Este resultado aproximado foi obtido após 33 iterações e com tempo de CPU de 1,07 segundos. A Figura 5.20 apresenta o comportamento da função objetivo do problema de otimização.

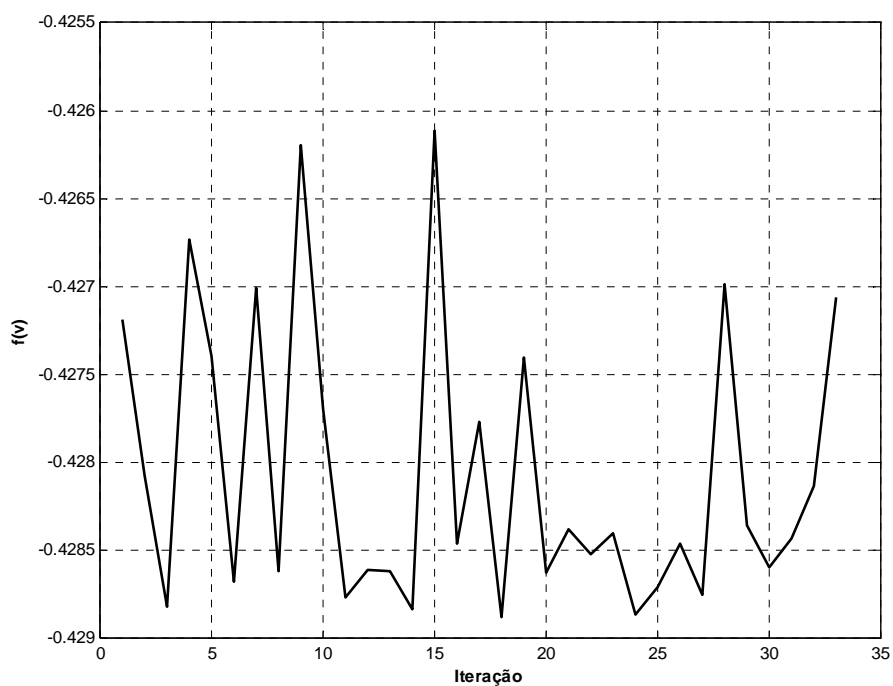


Figura 5.20: Comportamento da função objetivo.

5.2 Experimentos Baseados em Computação Evolutiva

Em Koziel e Michalewicz (1999) são propostos alguns *benchmarks* para problemas de otimização não-linear restrita. Nesta seção, um destes *benchmarks* é apresentado, por ser bastante utilizado como estudo de caso na literatura. A Rede de Hopfield Genética é comparada com outras abordagens (também citadas no Capítulo 2) que se utilizam da Computação Evolutiva para resolver tal problema.

Experimento: Seja o problema de otimização não-linear com restrições de desigualdade e variáveis canalizadas, cuja solução ótima é dada por $v^* = [14,095 \ 0,4296]^T$ e $f(v^*) = -7961,8$:

$$\begin{aligned} \text{Min } f(v) &= (v_1 - 10)^3 + (v_2 - 20)^3 \\ \text{Sujeito a: } & (v_1 - 5)^2 + (v_2 - 5)^2 - 100 \geq 0 \\ & -(v_1 - 6)^2 - (v_2 - 5)^2 + 82,81 \geq 0 \\ & 13 \leq v_1 \leq 100 \\ & 0 \leq v_2 \leq 100 \end{aligned}$$

A solução encontrada pela Rede de Hopfield Genética para este problema foi $v = [13,6053 \ 0]^T$ e $f(v) = -7953,1387$, com 167 iterações e tempo de CPU igual a 3,68 segundos. A Figura 5.21 e a Figura 5.22 apresentam graficamente o comportamento de v e $f(v)$, respectivamente.

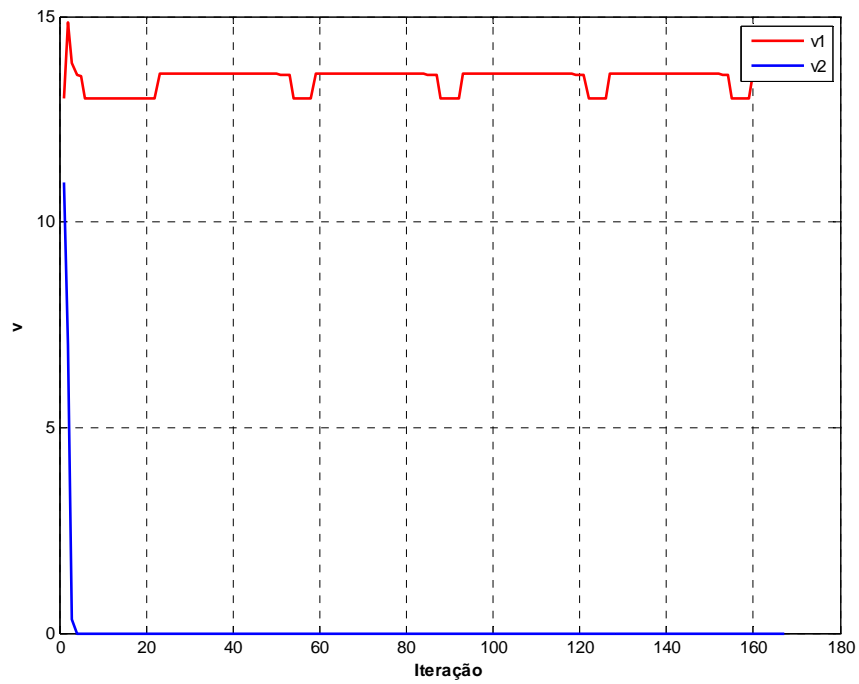


Figura 5.21: Evolução do vetor de saída da RHG.

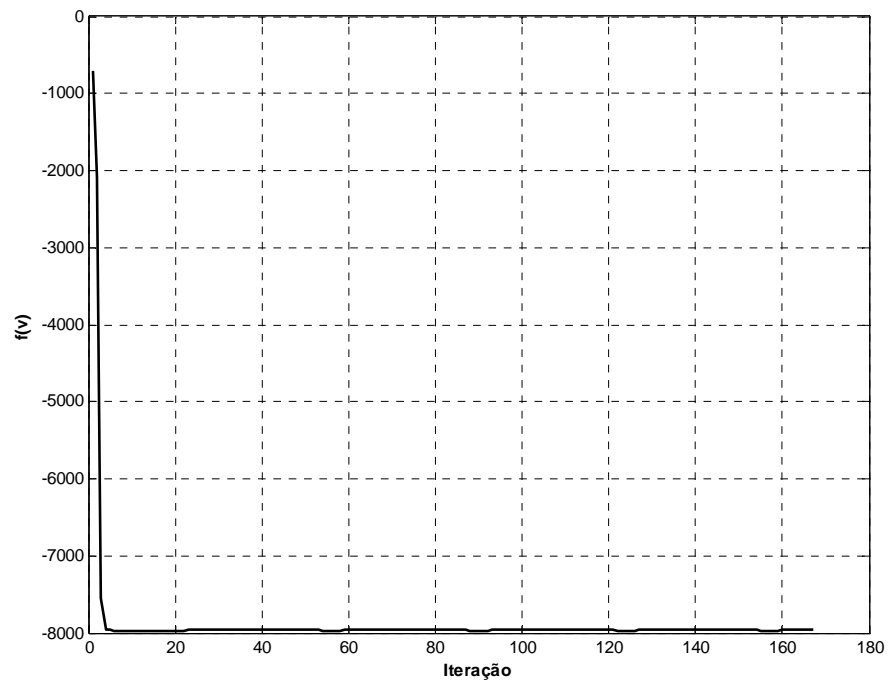


Figura 5.22: Comportamento da função objetivo.

Em seus experimentos, Koziel e Michalewicz (1999) encontraram valores para $f(v)$ que variam entre -7390,6 e -7961,1, sendo dependentes da definição de um parâmetro inicial. O método exige concentração de esforço computacional para delimitar a região factível de soluções e apresenta falhas em resolver problemas não-convexos.

Este problema também foi resolvido Deb (2000), Ming et al. (2003), Venkatraman e Yen (2005) e por Wu et al (2006). Deb (2000) através de um algoritmo genético associado a uma estratégia de penalidade, encontrou um valor aproximado, com $f(v) = -7840,8$. No entanto, o método proposto é falho quando o problema apresenta espaços de busca factíveis desconexos.

Ming et al. (2003) apresenta um algoritmo evolutivo com mecanismo de seleção de duas etapas, o qual tem sua eficiência prejudicada por utilizar parâmetros de penalidade, apesar de encontrar a solução bem próxima da ótima, com $f(v) = -7961,78$.

Um *framework* de duas fases baseado em algoritmos genéticos é proposto em Venkatraman e Yen (2005). O resultado obtido na resolução deste problema foi $f(v) = -7961,1785$. Entretanto, os autores relatam que a abordagem nem sempre encontra uma solução ótima ou próxima da ótima.

No trabalho de Wu et al. (2006), a proposição de um algoritmo genético multipopulação tenta reduzir o tempo de convergência gasto na obtenção de parâmetros de penalidade. O resultado obtido por eles foi $f(v) = -7910,7$. Os autores não relatam número de iterações ou tempo de processamento.

5.3 Problemas com Regiões de Busca Factíveis Desconexas

Em todos os experimentos apresentados nas Seções 5.1 e 5.2, havia apenas uma única região de busca factível, ou seja, um único espaço onde a RHG deveria buscar a solução.

Entretanto, existem problemas que possuem mais de uma região factível de busca, sendo que estas regiões estão desconectadas (espaços de busca disjuntos). O Experimento 1 a seguir, demonstra um problema deste tipo.

5.3.1 Experimento 1

Considere o problema de otimização apresentado em Koziel e Michalewicz (1999), cuja solução ótima corresponde a $v^* = [5 \ 5 \ 5]^T$ e $f(v^*) = -1$:

$$\text{Min } f(v) = (100 - (v_1 - 5)^2 - (v_2 - 5)^2 - (v_3 - 5)^2)/100$$

$$\text{Sujeito a: } (v_1 - p)^2 + (v_2 - q)^2 + (v_3 - r)^2 \leq 0.25, p, q, r = 1, 3, 5, 7, 9$$

$$0 \leq v_i \leq 10 \ (1 \leq i \leq 3)$$

No trabalho de Koziel e Michalewicz (1999) são consideradas $5^3 = 125$ espaços (esferas) de busca disjuntos, considerando as 125 combinações de p , q e r . Para a realização do experimento com a Rede de Hopfield Genética foram utilizadas apenas 5 combinações de p , q e r , gerando 5 regiões factíveis de busca. O resultado ótimo foi obtido após 3 iterações, com tempo de CPU de 0,27 segundo.

Mesmo considerando 125 regiões de busca disjuntas, Koziel e Michalewicz (1999) também encontraram a solução ótima em todas as suas execuções, relatando que a abordagem desenvolvida por eles não teve problemas em encontrar o ótimo global.

5.3.2 Experimento 2: Estudo de Caso em Identificação Robusta para Modelos Não-lineares

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 obtidas sobre o respectivo sistema. Estas informações são frequentemente afetadas pela

presença de incertezas que representam algum tipo de perturbação atuando no sistema observado.

Os métodos mais recentes de estimação paramétrica robusta calculam uma região (a menor possível) do espaço paramétrico (região de pertinência paramétrica) que é compatível com a estrutura do modelo em questão, 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) de valores possíveis para cada parâmetro estimado. Este intervalo é obtido a partir de valores assumidos por cada estimador na região de pertinência paramétrica.

Geralmente, uma aproximação simplificada para a região de pertinência paramétrica é obtida por meio de uma região hipercúbica mínima que seja externa à região de pertinência paramétrica e esteja alinhada com os eixos paramétricos (Belforte et al. 1988).

Em particular, a região hipercúbica mínima contendo a região de pertinência paramétrica 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 em análise.

O intervalo de incerteza paramétrica, correspondente ao i -ésimo parâmetro desconhecido, é definido por $[\theta_i^m, \theta_i^M]$, $i \in [1..N]$, sendo que N é a quantidade de parâmetros desconhecidos e os parâmetros θ_i^m e θ_i^M (pertencentes à região de pertinência paramétrica) são dados por:

$$\begin{aligned} \theta_i^m &= \min \theta_i, & i \in [1..N] \\ \theta_i^M &= \max \theta_i, & i \in [1..N] \end{aligned} \tag{5.1}$$

Finalmente, o estimador central θ^c é calculado por:

$$\theta^c = \frac{(\theta_i^m + \theta_i^M)}{2}, \quad i \in [1..N] \quad (5.2)$$

Estudo de Caso: Considere o problema de identificação robusta descrito por meio do modelo apresentado em (5.3):

$$y(k) = \begin{cases} \theta_2 - \theta_1 - \text{sen}(\theta_1) + e(k) & , \text{ se } k = 1..10 \\ \theta_2 - \theta_1 - \text{sen}(\theta_1 + 1) + e(k) & , \text{ se } k = 10..20 \end{cases} \quad (5.3)$$

sendo que $y(k)$ 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 $|e(k)| \leq 1$.

A Tabela 5.3 fornece as medidas simuladas para o exemplo.

Tabela 5.3: Vetor de medidas para o modelo não-linear.

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

Neste problema de identificação não-linear, a RHG, combinada com a perspectiva eixos-direcionados (Silva, 1995), a qual consiste em percorrer o espaço de busca em direção direta e oposta ao eixo cartesiano em relação a cada parâmetro de estimação, foi utilizada para calcular os intervalos de incerteza paramétrica e, conseqüentemente, obter o estimador central do modelo. O espaço paramétrico inicial para este problema, o qual pode ser implementado por meio da função de ativação rampa-simétrica, é dado por:

$$\Omega = \begin{cases} 0 \leq \theta_1 \leq 12 \\ 1 \leq \theta_2 \leq 11 \end{cases} \quad (5.4)$$

Na Figura 5.23, as regiões demarcadas representam os conjuntos de pertinência paramétrica exatos para o problema de identificação robusta definida pela equação (5.3).

Nota-se, para este exemplo, que a região de pertinência paramétrica é formada por duas sub-regiões desconectadas.

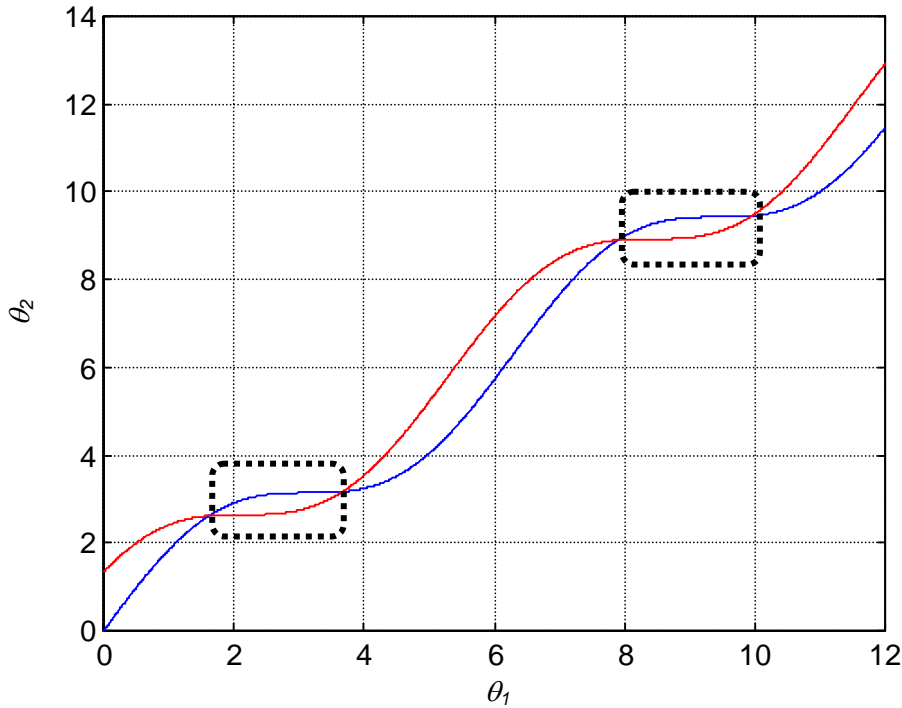


Figura 5.23: Região de pertinência paramétrica (modelo não-linear).

Para obter os intervalos de incerteza paramétrica, correspondentes à primeira sub-região, o espaço paramétrico inicial foi reduzido para $\{1 \leq \theta_1 \leq 6\}$, deixando θ_2 restrito ao domínio especificado para θ_1 .

Assim, os valores finais dos parâmetros θ_1 e θ_2 obtidos a partir da aplicação da RHG, correspondentes à primeira sub-região e considerando a aplicação da perspectiva eixos-direcionados, são dados conforme se apresenta na Tabela 5.4.

Tabela 5.4: Resultados do problema de estimação robusta (1ª sub-região).

	Valor de θ_1 após a convergência	Valor de θ_2 após a convergência
$f(v) = f(\theta) = -\theta_1$	3,6210	3,3426
$f(v) = f(\theta) = \theta_1$	1,5239	2,5488
$f(v) = f(\theta) = -\theta_2$	3,5348	3,3137
$f(v) = f(\theta) = \theta_2$	2,9076	2,5993

Na Tabela 5.5, observa-se os intervalos de incerteza paramétrica calculados pela abordagem proposta, considerando a Tabela 5.3.

Tabela 5.5: Intervalos de incerteza paramétrica obtidos pela RHG.

	Valor Mínimo (θ_i^m)	Valor Máximo (θ_i^M)
θ_1	1,5239	3,6210
θ_2	2,5488	3,3426

Para obter os intervalos de incerteza paramétrica, correspondentes à segunda sub-região, o espaço paramétrico inicial foi reduzido para $\{6 \leq \theta_1 \leq 12\}$, deixando θ_2 restrito ao domínio especificado para θ_1 . Os valores finais dos parâmetros θ_1 e θ_2 obtidos pela Rede de Hopfield Genética, considerando a segunda sub-região, são fornecidos na Tabela 5.6.

Tabela 5.6: Resultados do problema de estimação robusta (2ª sub-região).

	Valor de θ_1 após a convergência	Valor de θ_2 após a convergência
$f(v) = f(\theta) = -\theta_1$	9,8426	9,9892
$f(v) = f(\theta) = \theta_1$	7,7951	7,8482
$f(v) = f(\theta) = -\theta_2$	8,7905	9,1866
$f(v) = f(\theta) = \theta_2$	7,8873	8,6527

Com base nos valores apresentados na Tabela 5.6, obtêm-se então os valores de seus intervalos de incerteza paramétrica (Tabela 5.7), ou seja:

Tabela 5.7: Intervalos de incerteza paramétrica obtidos pela RHG.

	Valor Mínimo (θ_i^m)	Valor Máximo (θ_i^M)
θ_1	7,7951	9,8426
θ_2	7,8482	9,9892

Finalmente, os estimadores centrais, para cada sub-região, calculados a partir do ponto médio dos respectivos intervalos de incerteza paramétrica (definido na equação (5.2)), são fornecidos na Tabela 5.8.

Tabela 5.8: Estimadores centrais (modelo não-linear).

Estimador Central	1 ^a sub-região	2 ^a sub-região
θ_1^c	2,5724	8,8188
θ_2^c	2,9457	8,9187

Vale ressaltar que quando não há informação sobre as sub-regiões que porventura compõem a região de pertinência paramétrica, a RHG sempre convergirá para a sub-região que esteja mais próxima ao ponto onde a solução foi inicializada.

Esta forma, mostra-se também a eficiência da aplicação da RGH quando a mesma é utilizada na resolução de problemas de estimação paramétrica robusta.

5.4 Considerações Finais

Neste capítulo, a arquitetura neuro-genética foi aplicada na solução de alguns problemas de otimização não-linear restrita, assim como em um estudo de caso em identificação robusta para modelos não-lineares restritos. Os parâmetros internos da rede neural foram obtidos utilizando conceitos relativos às teorias de otimização não-linear e de estabilidade de Lyapunov. Todas as restrições, sejam de igualdade ou desigualdade, associadas aos problemas de otimização foram mapeadas através da técnica do subespaço-válido de soluções. A função objetivo foi otimizada através do algoritmo genético, utilizando os métodos e parâmetros apresentados no Capítulo 4.

Os experimentos mostraram que a Rede de Hopfield Genética desenvolvida fornece resultados bem precisos para a grande maioria dos problemas, em tempo de execução satisfatório, para os diversos tipos de problemas não-lineares restritos, confirmando-a como uma alternativa para solução de tais problemas. Além disso, a RHG sempre satisfaz todas as restrições estruturais impostas pelos referidos problemas.

CAPÍTULO 6

Conclusões e Trabalhos Futuros

6.1 Conclusões

Este trabalho se concentrou na aplicação de uma arquitetura híbrida, baseada nas metodologias de redes neurais e algoritmos genéticos, na solução de problemas de otimização não-linear restrita.

Os problemas utilizados nos experimentos visando propósitos de validação, mapeados e solucionados através da Rede de Hopfield Genética, foram também comparados com outros métodos inteligentes utilizados em suas soluções. As diversas simulações realizadas confirmam que a abordagem proposta é uma alternativa viável para solucionar problemas de otimização não-linear restrita, considerando tanto a precisão das soluções obtidas como também o tempo de convergência despendido para o alcance dos pontos de equilíbrio que representam tais soluções.

Com base nos experimentos, tornou-se possível verificar que a utilização da abordagem neuro-genética proporciona diversos benefícios quando comparada com os métodos primais (Bazaraa e Shetty, 1979), tradicionalmente utilizados na solução destes problemas. Dentre estes benefícios, destacam-se:

- (i) Ausência de necessidade de determinação, em cada iteração, do conjunto ativo de restrições.
- (ii) Em cada iteração, não é preciso calcular uma direção admissível de busca.

- (iii) A solução inicial não necessariamente precisa pertencer ao conjunto factível definido pelas restrições.
- (iv) Implementação direta das variáveis canalizadas através da função de ativação dos neurônios da rede de Hopfield.
- (v) Ausência de uso de multiplicadores de Lagrange.

Quando comparada a uma grande parcela dos métodos inteligentes usados na solução de problemas de otimização restrita, a arquitetura neuro-genética proposta também possui os seguintes benefícios:

- (i) Mapeamento de problemas de otimização não-linear de naturezas diferentes, utilizando uma mesma metodologia, independentemente do problema tratado.
- (ii) Capacidade de resolver os diferentes tipos de problemas de otimização não-linear restrita, demonstrando eficiência significativa quando comparada com outros métodos disponíveis na literatura correlata.
- (iii) Maior simplicidade computacional para mapear problemas.
- (iv) Viabilização do tratamento das restrições envolvidas com os problemas.
- (v) Ausência de necessidade de definição de parâmetros de controle ou ponderação para sua inicialização.
- (vi) Possibilidade de implementação em hardware.

Tais benefícios comprovam que a sintetização de uma estrutura neuro-genética capaz de solucionar problemas de otimização não-linear é uma ferramenta poderosa e vantajosa, além de fornecer um método inovador para solução deste tipo de problema. Os ganhos advindos da simplicidade relativa aos aspectos computacionais e da aplicabilidade da estrutura proposta corroboram a utilização da arquitetura desenvolvida neste trabalho.

6.2 Trabalhos Futuros

Para alguns dos experimentos realizados, observou-se que a Rede de Hopfield Genética não alcançou a solução ótima ou gastou tempo demais para encontrar uma solução aceitável. Assim, propõe-se como trabalho futuro um maior refinamento da abordagem neuro-genética, melhorando ainda mais sua precisão e reduzindo seu tempo de convergência.

Outra proposição de trabalho futuro seria a de desenvolver um algoritmo genético multi-objetivo para otimização não-linear restrita, proporcionando avaliar restrições e função objetivo através de uma abordagem evolutiva. Em seguida, seriam realizadas comparações entre as soluções obtidas pelo algoritmo genético multi-objetivo e as soluções encontradas pela Rede de Hopfield Genética.

Por fim, pretende-se também utilizar a RHG em aplicações a serem desenvolvidas junto ao Grupo de Pesquisa Básica e Avançada em Sistemas Inteligentes (GPBASI), do curso de Engenharia da Computação da Universidade Estadual de Feira de Santana, onde a autora é professora e pesquisadora.

Referências Bibliográficas

- Ai, W., Song, Y. e Chen, Y. (2006). An improved neural network for solving optimization of quadratic programming problems. **Proceedings of the Fifth International Conference on Machine Learning and Cybernetics**, pp. 3083-3088.
- Aiyer, S. V. B., Niranja, M. e Fallside, F. (1990). A theoretical investigation into the performance of the Hopfield model. **IEEE Transactions on Neural Networks**, v. 1, n. 2, pp. 204-215.
- Andino, A., Araujo, L., Sáenz, F. e Ruz, J. (2000). A hybrid evolutionary approach for solving constrained optimization problems over finite domains. **IEEE Transactions on Evolutionary Computation**, v. 4, n. 4, pp. 353-372.
- Bazaraa, M. S. e Shetty, C. M. (1979). **Nonlinear Programming: Theory and Algorithms**, John Wiley & Sons.
- Bazaraa, M. S., Sherali, H. D. e Shetty, C. M. (1993). **Nonlinear Programming: Theory and Algorithms (Second Edition)**, John Wiley & Sons, Inc.
- Beasley, D., Bull, D. R. e Martin, R. R. (1993). **An overview of genetic algorithms: Part 1, Fundamentals**. University Computing. Disponível em: <http://surf.de.uu.net/encore/GA/papers/over93.ps.gz>. Acesso em: 08/08/2007.
- Belforte, G., Bona, B. e Cerone, V. (1988). **Parameter estimation with set membership uncertainty: Nonlinear families of models**. Proceedings of the 8th Symposium on Identification and System Parameter Estimation, Beijing, pp. 399-404.

- Bouzerdoun, A. e Pattison, T. R. (1993). Neural network for quadratic optimization with bound constraints. **IEEE Transactions on Neural Networks**, v. 4, n. 2, pp. 293-304.
- Braga, A. P., Ludermir, T. B. e Carvalho, A. C. P. L. F. (2000). **Redes Neurais Artificiais - Teoria e Aplicações**, LTC.
- D'Azzo, J. J. e Houpis, C. H. (1975). **Linear Control System Analysis and Design**, McGraw-Hill.
- Deb, K. (2000). An efficient constraint handling method for genetic algorithm. **Computer Methods and in Applied Mechanics and Engineering**, v. 186, pp. 311-338.
- Delgado, M. R. (2002). **Projeto Automático de Sistemas Nebulosos: Uma Abordagem Co-Evolutiva**. (Tese de Doutorado). Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas-São Paulo.
- Fang, Y., e Kincaid, T. G. (1996). Stability analysis of dynamical neural networks. **IEEE Transactions on Neural Networks**, v. 7, n. 4, pp. 996-1006.
- Gee, A. H. e Prager, R. W. (1991). **Alternative Energy Functions for Optimizing Neural Networks**. Cambridge University Department of Engineering, Technical Report Cued/F-Infeng/TR95.
- Hao, X., Gao, H., Sun, C. e Liu, B. (2006). A model solving constrained optimization problem based on the stability of hopfield neural network. **Proceedings of the Sixth World Congress on Intelligent Control and Automation**, pp. 2790- 2795.
- Haykin, S. (1999). **Neural Networks - A Comprehensive Foundation**, Prentice-Hall, New York.

- Hegde, S. U., Sweet, J. L. e Levy, W. B. (1988). Determination of parameters in a Hopfield/Tank computational network. **Proceedings of the International Conference on Neural Network**, v. 2, n. 1, pp. 291-298.
- Hodgkin, A. L. e Huxley, L. F. (1952). A quantitative description of membrane current and its applications to conduction and excitation in nerve. **Journal of Physiology**, v. 117, pp. 500-544.
- Holland, J. H. (1975). **Adaptation in Natural and Artificial Systems**, Ann Arbor: The University of Michigan Press.
- Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. **Proceedings of the National Academy USA**, v. 81, n. 1, pp. 3088-3092.
- Hopfield, J. J. e Tank, D. W. (1985). Neural computation of decisions in optimization problems. **Biological Cybernetics**, v. 52, pp. 141-152.
- Hu, X. e Wang, J. (2006). A recurrent neural network for solving nonconvex optimization problems. **IEEE International Joint Conference on Neural Networks**, pp. 8955-8961.
- Hush, D. R. e Horne, B. G. (1993). Progress in supervised neural networks. **IEEE Signal Processing Magazine**, v. 81, pp. 3088-3092.
- Jang, J. -S. R., Sun, C. -T, Mizutani, E. (1997). **Neuro-Fuzzy and Soft Computing**, Prentice Hall, Englewood Cliffs.
- Kamgar-Parsi, B. e Kamgar-Parsi, B. (1990). On problem solving with Hopfield neural networks. **Biological Cybernetics**, v. 62, pp. 415-423.

- Kennedy, M. P. e Chua, L. O. (1988). Neural networks for nonlinear programming. **IEEE Transactions on Circuits and Systems**, v. 35, n. 5, pp. 554-562.
- Koziel, S. e Michalewicz, Z. (1999). Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. **Evolutionary Computing**, vol. 7(1), pp. 19–44.
- Lacerda, E. G. M. (2003). **Model Selection of RBF Networks via Genetic Algorithms**. (Tese de Doutorado). Departamento de Ciências da Computação, Universidade Federal de Pernambuco.
- Lacerda, E. G. M. e Carvalho, A. C. P. L. F. (1999). Introdução aos Algoritmos Genéticos. **Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação**, v. 2, pp. 51-126.
- Le, T. V. (1996). A fuzzy evolutionary approach to constrained optimisation problems. **Proceedings of the IEEE International Conference on Evolutionary Computation**, pp. 274 - 278.
- Liang, X. e Wang, J. (2000). A recurrent neural network for nonlinear optimization with a continuously differentiable objective function and bound constraints. **IEEE Transactions on Neural Networks**, v. 11, n. 6, pp. 1251-1262.
- Lua, B. e Itob, K. (2003). Converting general nonlinear programming problems into separable programming problems with feedforward neural networks. **The Official Journal of the International Neural Network Society**, v. 16, pp. 1059 -1074
- Luenberger, D. G. (1984). **Linear and Nonlinear Programming**, Addison Wesley.
- Matlab (2006). Matlab Genetic algorithm and direct search toolbox user's guide. © COPYRIGHT 2004–2006 by The MathWorks, Inc.

- McCulloch, W. S. e Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, pp. 115-133.
- Michalewicz, Z. (1996). Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag.
- Ming, C., Kamhiro, O., Kanji, U. e Masahara, S. (2003). A two-step selection scheme for constrained evolutionary optimization. **IEEE International Conference on Neural Networks & Signal Processing**, v. 1, pp. 14-17.
- Mitchell, M. (1996). An Introduction to Genetic Algorithms, MIT Press.
- Montes, E. e Coelho, C. (2005). A simple multimembered evolution strategy to solve constrained optimization problems. **IEEE Transactions on Evolutionary Computation**, v. 9, n. 1, pp. 1-17.
- Osório, F. e Vieira, R. (2003). **Inteligência Artificial e Sistemas Inteligentes**. Material de aula do Programa Interdisciplinar de Pós-Graduação - Mestrado em Computação Aplicada da Unisinos, Unisinos, São Leopoldo-RS.
- Peterson, C. e Soderberg, B. (1989). A new method for mapping optimization problems onto neural networks. **International Journal of Neural Networks**, v. 1, pp. 3-22.
- Rezende, S. O. (2003). **Sistemas Inteligentes: Fundamentos e Aplicações**, Editora Manoele, 1ª edição.
- Rodríguez-Vázquez, A., Domínguez-Castro, R., Rueda, A., Huertas, J. L. e Sánchez-Sinencio, E. (1990). Nonlinear switched-capacitor neural networks for optimization problems. **IEEE Transactions on Circuits and Systems**, v. 37, n. 1, pp. 384-397.
- Rosenbrock, H. H. e Storey, C. (1970). **Mathematics of Dynamical Systems**, Thomas Nelson & Sons.

- Silva, I. N. (1995). **Estimação Paramétrica Robusta Através de Redes Neurais Artificiais**. (Dissertação de Mestrado). DCA/FEE, Universidade de Campinas, Campinas-São Paulo.
- Silva, I. N. (1997). **Uma Abordagem Neuro-Nebulosa Para Otimização de Sistemas e Identificação Robusta**. (Tese de Doutorado). DCA/FEE, Universidade de Campinas, Campinas-São Paulo.
- Tank D. W. e Hopfield J. J. (1986). Simple neural optimization networks: an A/D converter, signal decision network, and a linear programming circuit. **IEEE Transactions on Circuits and Systems**, vol. CAS-33, pp. 533-545.
- Venkatraman, S. e Yen, G. (2005). A generic framework for constrained optimization using genetic algorithms. **IEEE Transactions on Evolutionary Computation**, v. 9, n. 4, pp. 424-435.
- Vidyasagar, M. (1993a). Location and stability of the high-gain equilibria of nonlinear neural networks. **IEEE Transactions on Neural Networks**, v. 4, n. 4, pp. 660 - 672.
- _____(1993b). **Nonlinear Systems Analysis (Second Edition)**, Prentice Hall, Englewood Cliffs.
- Wilson, V. e Pawley, G. S. (1988). On the stability of the TSP problem algorithm of Hopfield and Tank. **Biological Cybernetics**, v. 58, pp. 63-70.
- Wu, Y., Lu, J. e Sun, Y. (2006). An improved multi-population genetic algorithm for constrained nonlinear optimization. **Proceedings of the Sixth World Congress on Intelligent Control and Automation**, v. 1, pp. 1910-1914.
- Xia. Y., Leung, H. e Wang, J. (2002). A projection neural network and its application to constrained optimization problems. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, v. 49, n. 4, pp. 447-458.

- Xia, Y. e Wang, J. (2003). A recurrent neural network for nonlinear convex programming. **Proceedings of the International Symposium on Circuits and Systems**, v. 3, n. 1, pp. 470-473.
- Xia, Y. e Wang, J. (2004). A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints **IEEE Transactions on Circuits and Systems - Part I : Regular Papers** v. 51, n. 7, pp. 1385-1394.
- Xia, Y. e Feng, G. (2007). A new neural network for solving nonlinear projection equations. **The Official Journal of the International Neural Network Society** (*Aceito para publicação*).
- Zak, S. H., Upatising, V. e Hui, S. (1995). Solving linear programming problems with neural networks: a comparative study. **IEEE Transactions on Neural Networks**, v. 6, n. 1, pp. 94-104.
- Zheng, Y., Ma, L. e Qian, J. (2002). Neural network solution for general nonlinear optimization problems. **TENCON: IEEE Conference on Computers, Communications, Control and Power Engineering**, v. 3, n. 1, pp. 1713-1716.

Apêndice A

Implementação em Hardware da Rede Neural de Hopfield

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) é excitatória ($T_{ij} > 0$), este resistor é conectado a saída normal (+) do amplificador j . Para uma sinapse inibitória ($T_{ij} < 0$), é conectada a saída inversa (-) do amplificador j . A matriz 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. As equações que descrevem a evolução deste circuito em relação ao tempo são definidas por:

$$C_i \left(\frac{du_i}{dt} \right) = \sum_{j=1}^N T_{ij} V_j - \frac{u_i}{R_i} + I_i \quad (\text{A.1})$$

sendo que $V_j = g_j(u_j)$.

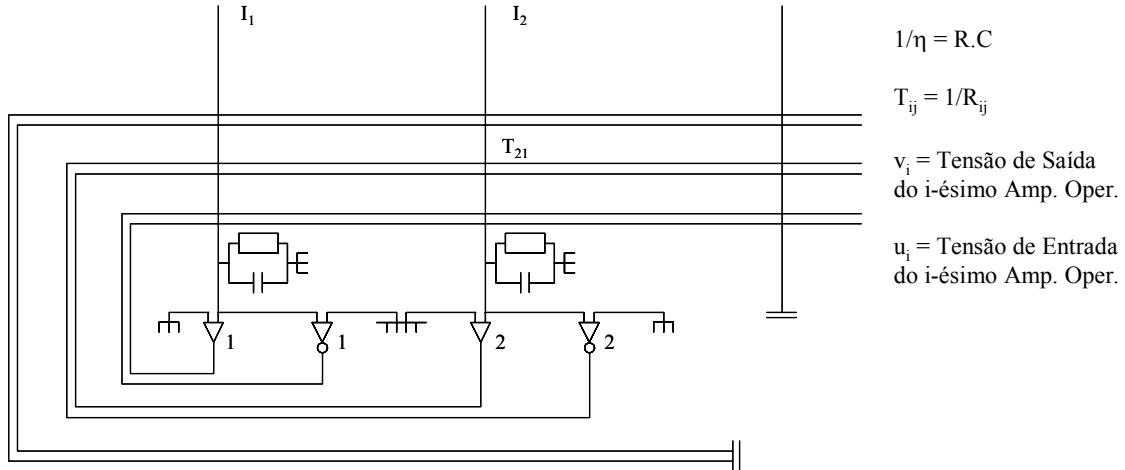


Figura A. 1: Hardware Analógico da Rede de Hopfield.

Define-se o termo R_i como a associação em paralelo de R_i^{in} e os termos R_{ij} , ou seja:

$$\frac{1}{R_i} = \frac{1}{R_i^{in}} + \sum_{j=1}^N \left(\frac{1}{R_{ij}} \right) \quad (\text{A.2})$$

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:

$$\frac{du_i}{dt} = -\eta u_i + \sum_{j=1}^N T_{ij} V_j + I_i \quad (\text{A.3})$$

sendo que $\eta = 1 / RC$ e $V_j = g(u_j)$.

Vale notar que a Equação (A.3) é idêntica a equação (3.1). A Equação (A.3) fornece uma descrição completa da evolução temporal do estado do circuito. A integração desta em computador digital permite a simulação de qualquer configuração hipotética da rede de Hopfield.

Apêndice B

Algoritmo Genético com Codificação Binária

Como mencionado no Capítulo 4, a Rede de Hopfield Genética inicialmente utilizada no desenvolvimento deste trabalho, era composta por um algoritmo genético com codificação binária, o qual apresentou alguns problemas. A perda de precisão em função de conversão numérica entre bases (real, decimal e binária) foi a mais freqüente, fazendo com que a RHG gastasse mais tempo para encontrar uma precisão aceitável. Além disso, tentativas de inserir melhorias são mais difíceis de serem realizadas em operadores genéticos para codificação binária, uma vez que estes realizam apenas trocas de bits ou de conjuntos de bits.

Para exemplificar o problema, a Figura A. 2 e a Figura A. 3 ilustram, respectivamente, a evolução do vetor v de saída da RHG e o comportamento da função objetivo do problema de otimização não-linear restrita, composto por uma função objetivo não-convexa, restrições de desigualdade e variáveis canalizadas, mostrado no Experimento 3 (Capítulo 5).

Este problema tem única solução ótima dada por $v^* = [0,0000 \ 1,5000 \ 0,0000]^T$, com função objetivo $f(v^*) = -3,5000$.

O resultado obtido pela RHG com codificação binária foi $v = [0,0000 \ 1,5451 \ 0,0098]^T$ e $f(v) = -3,4764$, executando 1191 iterações e gastando um tempo de CPU de aproximadamente 1,4 minutos. Este resultado foi pior do que o encontrado pela RHG com codificação real, o qual vale $v = [0,0000 \ 1,5022 \ 0,0005]^T$ e $f(v) = -3,4991$, com tempo de CPU de 29,54 segundos e 547 iterações.

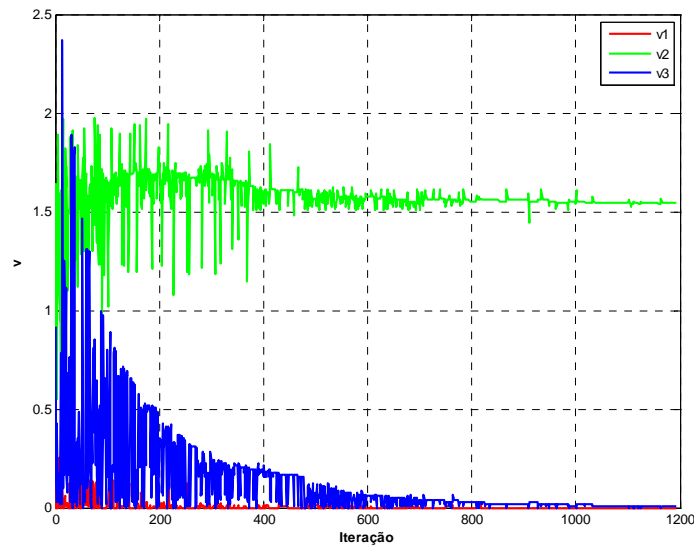


Figura A. 2: Evolução de v - algoritmo genético (codificação binária).

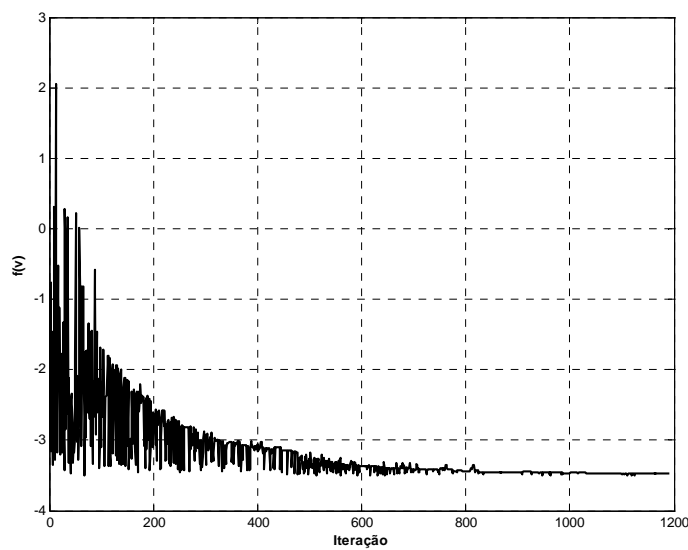


Figura A. 3: Evolução de $f(v)$ - algoritmo genético (codificação binária).

Outra observação importante é que para problemas com muitas variáveis (Experimento 7, Capítulo 5), os cromossomos se tornavam muito grandes, tornando o processo de busca mais lento e diminuindo ainda mais a eficiência do método.