

Oscar Javier Begambre Carrillo

**Algoritmo Híbrido para Avaliação da Integridade Estrutural: uma
Abordagem Heurística**

Tese apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo como requisito para obtenção do título de Doutor em Engenharia Civil.

Área de concentração: Estruturas
Orientador: Professor Titular José Elias Laier

São Carlos
2007

Ao meu filho Sergio.
A minha esposa Liliana.

AGRADECIMENTOS

Eu tive a sorte de realizar esta pesquisa com um excelente grupo de professores, funcionários e alunos de pós-graduação no Departamento de Estruturas da Escola de Engenharia de São Carlos da Universidade de São Paulo-USP desde 2004.

Gostaria de agradecer, de forma especial, ao Professor Titular Dr. José E. Laier, por ter me dado a oportunidade de trabalhar com ele e por sua competente orientação nestes últimos três anos.

Aos meus amigos, os professores Dr. Wilson Venturini, Dr. Samuel Giongo, Dr. Humberto Coda, Dra. Heloisa Sobreiro pelas longas e valiosas horas de conversa, a Carlos Azevedo, Odília de Azevedo, Wesley Góis, Leandro Waidemam, Walter Oliveira, Claudius Barbosa, Priscila Castilho, Neire Castilho e Sandra Almeida, que sempre estiveram comigo durante minha permanência em São Carlos.

A Rosi Lopes e a Maria Nadir Minatel, pela amizade e as orientações fornecidas para a elaboração deste documento.

Ao CNPq pela bolsa de estudos, sem a qual não teria conseguido enfrentar o desafio de escrever esta tese.

À minha família.

Enquanto não alcançares a verdade, não poderás corrigi-la. Porém, se a não corrigires, não a alcançarás. Entretanto, não te resignes.

Do Livro dos Conselhos.

RESUMO

BEGAMBRE, O. Algoritmo Híbrido para Avaliação da Integridade Estrutural: Uma Abordagem Heurística. 2007. 152 f. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.

Neste estudo, o novo algoritmo híbrido autoconfigurado PSOS (*Particle Swarm Optimization – Simplex*) para avaliação da integridade estrutural a partir de respostas dinâmicas é apresentado. A formulação da função objetivo para o problema de minimização definido emprega Funções de Resposta em Frequência e/ou dados modais do sistema. Uma nova estratégia para o controle dos parâmetros do algoritmo *Particle Swarm Optimization* (PSO), baseada no uso do método de Nelder – Mead é desenvolvida; conseqüentemente, a convergência do PSO fica independente dos parâmetros heurísticos e sua estabilidade e precisão são melhoradas. O método híbrido proposto teve melhor desempenho, nas diversas funções teste analisadas, quando comparado com os algoritmos *Simulated Annealing*, Algoritmos Genéticos e o PSO. São apresentados diversos problemas de detecção de dano, levando em conta os efeitos do ruído e da falta de dados experimentais. Em todos os casos, a posição e extensão do dano foram determinadas com sucesso. Finalmente, usando o PSOS, os parâmetros de um oscilador não linear (oscilador de Duffing) foram identificados.

Palavras-chave: *Particle Swarm Optimization*, *Simulated Annealing*, identificação de dano, problemas inversos, vigas fissuradas, oscilador não linear.

ABSTRACT

Begambre, O. Hybrid algorithm for damage detection: a heuristic approach. 2007. 152 f. Thesis (Doctoral). Escola de Engenharia de São Carlos, Universidade de São Paulo.

In this study, a new auto configured Particle Swarm Optimization – Simplex Algorithm for damage detection has been proposed. The formulation of the objective function for the minimization problem is based on the Frequency Response Functions (FRFs) and the modal parameters of the system. A novel strategy for the control of the Particle Swarm Optimization (PSO) parameters based on the Nelder–Mead algorithm (Simplex method) is presented; consequently, the convergence of the PSOS becomes independent of the heuristic constants and its stability and accuracy are enhanced. The formulated hybrid method performs better in different benchmark functions than the Simulated Annealing (SA), the Genetic Algorithm (GA) and the basic PSO. Several damage identification problems, taking into consideration the effects of noisy and incomplete data, were studied. In these cases, the damage location and extent were determined successfully. Finally, using the PSOS, a non-linear oscillator (Duffing oscillator) was identified with good results.

Keywords: Particle Swarm Optimization, Simulated Annealing, damage identification, inverse problems, cracked beam, non-linear oscillator.

LISTA DE ABREVIATURAS

| | |
|--------|--|
| AG | Algoritmo Genético |
| ARMA | <i>Autoregressive Moving Average</i> |
| COMAC | <i>Coordinate Modal Assurance Criterion</i> |
| DD | Detecção de Dano |
| FFT | Transformada Rápida de Fourier |
| FRAC | <i>Frequency Response Assurance Criterion</i> |
| FRF | Função de Resposta em Frequência |
| GAC | <i>Global Amplitude Criterion</i> |
| GSC | <i>Global Shape Criterion</i> |
| HFRF | <i>High Frequency Response Function</i> |
| MEF | Método dos Elementos Finitos |
| PD | Problema Direto |
| PI | Problema Inverso |
| PIP | Problema de Identificação Paramétrica |
| PNL | Programação Não Linear |
| PSO | <i>Particle Swarm Optimization</i> |
| RNA | Rede Neural Artificial |
| RNA-BP | <i>Rede Neural Artificial Back Propagation</i> |
| SA | <i>Simulated Annealing</i> |
| TBM | Técnicas Baseadas em Modelos |
| TBS | Técnicas Baseadas em Sinais |
| TW | Transformada Wavelet |

SUMÁRIO

| | |
|---|-----------|
| 1 INTRODUÇÃO | 1 |
| 1.1 Necessidade de Avaliação da Integridade Estrutural | 1 |
| 1.2 Métodos de Detecção de dano a partir de respostas dinâmicas | 4 |
| 1.3 Detecção de dano como Problema Inverso | 6 |
| 1.4 Objetivos da Tese | 8 |
| 1.5 Alcance | 9 |
| 1.6 Síntese do conteúdo da Tese | 10 |
| 2 DETECÇÃO DE DANO-ESTADO DA ARTE | 12 |
| 2.1 Introdução | 12 |
| 2.2 Técnicas Baseadas em Modelos (TBM) | 14 |
| 2.2.1 Métodos que empregam dados Modais | 17 |
| 2.2.2 Métodos que Empregam Dados no Domínio da Frequência | 21 |
| 2.2.3 Métodos que Empregam Dados no Domínio do Tempo | 23 |
| 2.3 Técnicas Baseadas em Sinais (TBS) | 24 |
| 2.4 Métodos Heurísticos | 26 |
| 3 DETECÇÃO DE DANO: BASES TEÓRICAS | 31 |
| 3.1 Introdução | 31 |
| 3.2 Correlação entre Resultados Teóricos e Experimentais | 34 |
| 3.3 Modelos de Dano | 35 |
| 3.4 Equação de Movimento via Método dos Elementos Finitos (MEF) | 38 |
| 3.4.1 Problema do Auto Valor para Vibração Livre Não Amortecida | 40 |
| 3.4.2 Amortecimento Proporcional | 42 |
| 3.4.3 Amortecimento Geral | 43 |
| 3.4.4 Funções de Resposta em Frequência Analíticas (FRFs) | 46 |
| 3.4.5 Introdução de Dano no MEF da Estrutura | 48 |
| 3.5 Uso de Dados Experimentais para Avaliação da Integridade Estrutural | 49 |
| 3.5.1 Funções Objetivo | 50 |
| 3.6 Análise de Resultados | 54 |
| 4 ALGORITMOS HEURÍSTICOS PARA DETECÇÃO DE DANO | 56 |

| | |
|---|------------|
| 4.1 Introdução | 56 |
| 4.2 Algoritmo <i>Simulated Annealing</i> (SA) | 59 |
| 4.2.1 Função Objetivo | 61 |
| 4.2.2 Estruturas de Vizinhança | 62 |
| 4.2.3 Programa de Esfriamento | 63 |
| 4.2.4 Critério de Parada | 64 |
| 4.3 Algoritmo <i>Particle Swarm Optimization</i> (PSO) | 66 |
| 4.3.1 Mecanismo e Parâmetros Básicos do PSO | 67 |
| 4.3.2 Topologias de Vizinhança | 70 |
| 4.3.3 Critérios de Parada | 71 |
| 4.4 Algoritmos Genéticos (AGs) | 72 |
| 4.4.1 Operadores e Parâmetros Básicos dos AGs | 73 |
| 4.4.2 Critérios de Parada | 75 |
| | |
| 5 O NOVO ALGORITMO HÍBRIDO: PSOS (PSO-SIMPLEX) | 76 |
| 5.1 Introdução | 76 |
| 5.2 Algoritmos Autoconfigurados | 77 |
| 5.3. O Algoritmo Nelder - Mead (N-M) | 78 |
| 5.4 O Algoritmo PSOS | 80 |
| 5.4.1 Heurística do PSOS | 81 |
| 5.4.2 Critérios de Parada do PSOS | 86 |
| 5.5 Análise de Convergência | 87 |
| | |
| 6 AVALIAÇÃO EXPERIMENTAL DOS ALGORITMOS | 89 |
| 6.1 Introdução | 89 |
| 6.2 Critérios de Avaliação dos Algoritmos: Precisão, Estabilidade, Robustez, Custo Computacional e Confiança | 89 |
| 6.3 Funções Teste - Resultados | 90 |
| | |
| 7 DETECÇÃO DE DANO USANDO OS ALGORITMOS PSOS E SA E DADOS MODAIS: EXEMPLOS NUMÉRICOS E ANALÍTICOS | 98 |
| 7.1 Introdução | 98 |
| 7.2 Problema de Kabe: Tolerância ao Ruído | 98 |
| 7.3 Identificação de Dano Viga em Balanço | 103 |
| | |
| 8 DETECÇÃO DE DANO USANDO O PSOS E FRFS: EXEMPLOS NUMÉRICOS E ANALÍTICOS | 107 |
| 8.1 Introdução | 107 |
| 8.2 Treliça de Dez Barras | 107 |

| | |
|---|------------|
| 8.3 Viga Livre-Livre com Fissuras | 110 |
| 8.4 Oscilador Não-Linear | 112 |
| 9 CONCLUSÕES, CONTRIBUIÇÕES E TRABALHO FUTURO | 115 |
| 9.1 Conclusões | 115 |
| 9.2 Contribuições | 116 |
| 9.3 Trabalho futuro | 118 |
| BIBLIOGRAFIA | 119 |
| APÊNDICE A: LISTAGEM PROGRAMA SIMULATED ANNEALING | 132 |
| APÊNDICE B: LISTAGEM DO PROGRAMA PARTICLE SWARM OPTIMIZATION | 139 |
| APÊNDICE C: LISTAGEM DO PROGRAMA PSOS | 145 |

1 Introdução

1.1 Necessidade de Avaliação da Integridade Estrutural

Avaliar a integridade estrutural utilizando técnicas baseadas na resposta dinâmica da estrutura tem se tornado, nas últimas décadas, um procedimento muito atraente para as indústrias de construção civil e aeroespacial, devido à possibilidade de detectar danos em sua infraestrutura de forma rápida e econômica, diminuindo desta forma, os riscos de perda de vidas humanas e de capital que o colapso de grandes estruturas pode acarretar.

Em termos gerais, dano pode ser definido como as variações introduzidas em um sistema, que afetam adversamente seu atual ou futuro desempenho. Implícito nesta definição, está o conceito de que dano não tem significado sem uma comparação entre dois diferentes estados do sistema, um dos quais é admitido como o inicial e quase sempre é o estado não danificado (FARRAR; DOEBLING; DUFFEY, 2000; DOEBLING; FARRAR; PRIME, 1998).

Os principais objetivos de um sistema de Detecção de Danos (DD) são: fornecer informação sobre o estado da estrutura, detectar potenciais estados de falha e permitir que sejam tomadas a tempo, as medidas necessárias para evitar a ruína do sistema estrutural.

Atualmente, existem condições tecnológicas, econômicas e legais que impulsionam o desenvolvimento e emprego de tecnologias para DD. Por exemplo, a despesa nos Estados Unidos com reparos de pontes obsoletas, custa anualmente aproximadamente 10 bilhões de dólares (CHASE, 2001). Hoje, a decisão sobre a saúde das pontes é feita através de inspeções visuais e utilizando métodos locais de detecção de dano, isto significa que muitas estruturas em condições péssimas de funcionamento podem não ser identificadas, aumentando o risco para os usuários. Por outro lado, pontes que não precisariam de reparo podem ser substituídas sem necessidade. Uma tecnologia de detecção de danos robusta pode ser utilizada, com grandes vantagens, para evitar estes erros.

A indústria do petróleo, e especialmente suas plataformas marítimas, também pode ser beneficiada pelo desenvolvimento de técnicas de inspeção *on-line* para sua infra-estrutura.

O desenvolvimento de uma nova geração de aeronaves, que satisfaça os crescentes requerimentos de desempenho da indústria aeroespacial, pode incorporar técnicas de diagnóstico estrutural automáticas e em tempo real, para abaixar custos operacionais e melhorar seus níveis de confiabilidade e segurança.

Motivados por razões de segurança, agências da Ásia estão exigindo das construtoras de grandes obras de infra-estrutura civil a instrumentação

das mesmas e a certificação periódica de sua “saúde estrutural” (FARRAR; DOEBLING; DUFFEY, 2000).

Infelizmente, no Brasil o desenvolvimento de tecnologias para DD não é uma prioridade do governo, porém, espera-se um aumento de interesse e de investimentos nesta área num futuro próximo.

Devido à necessidade de encontrar um método de detecção de danos em estruturas civis que se ajuste aos requerimentos e condições acima mencionados, a pesquisa nesta área se faz obrigatória, tendo em vista que, no futuro próximo, a indústria da construção civil deverá aplicar esta tecnologia para garantir a segurança da infra-estrutura por ela construída.

Hoje em dia, não existe uma técnica geral de detecção de danos em estruturas civis reconhecida pela comunidade científica. A seguir são citados alguns dos requerimentos para um sistema de avaliação da integridade estrutural, que possa ser utilizado em infra-estrutura civil.

-Precisão, estabilidade e robustez do procedimento de identificação de dano. Quando testado em diferentes sistemas estruturais, o método deve fornecer uma solução factível e chegar numa solução ótima ou quase ótima e deve ser tolerante ao ruído.

- A estrutura deve ser monitorada. Para isto, é necessário planejar uma distribuição adequada de sensores dentro da estrutura (KURATA; SPENCER; RUIZ-SANDOVAL, 2003). Se o controle for parte do sistema, a distribuição ótima de atuadores deverá ser levada em conta. O processamento de dados deve ser robusto e eficiente.

- Incorporar, nas estruturas a serem construídas, novos materiais que possam atuar como sensores/atuadores distribuídos e que permitam seu

automonitoramento e autocontrole. Este tipo de estruturas é conhecido como Estruturas Inteligentes (*Smart Structures*) (RAY; MALLIK, 2004).

- Combinar a DD com as técnicas de detecção local. Como as técnicas DD geralmente não fornecem informação suficiente para determinar a classe de dano, os métodos experimentais puros (locais) podem ser utilizados para identificar o tipo específico de dano presente, permitindo assim, definir uma estratégia de reparo ou substituição do elemento afetado.

As anteriores considerações constituem a motivação para a busca de novos métodos de detecção de dano baseados em modelos, como os apresentados nesta pesquisa. Neste trabalho, são avaliados dois métodos heurísticos (o SA e o PSO) para possível uso no problema DD, e é proposto um novo algoritmo heurístico híbrido para sua solução.

1.2 Métodos de Detecção de dano a partir de respostas dinâmicas

Os métodos de detecção de dano podem ser classificados em dois grandes grupos, de acordo com sua dependência (ou não) de um modelo estrutural, encontrando-se: os baseados em sinais (experimentais) e os baseados em modelos (ZOU; TONG; STEVEN, 2000). Por outro lado, Rytter (1993) define uma estrutura hierárquica de quatro níveis para o problema de detecção. A idéia fundamental do nível um ou nível de detecção (mais baixo) é determinar, com confiabilidade, se aconteceu ou não dano dentro da estrutura. Nos níveis intermediários, dois e três, busca-se determinar a posição e a extensão (severidade) da falha, respectivamente. No nível mais alto (nível quatro), espera-se que a técnica de detecção possa prever a vida

útil remanescente da estrutura, após a ocorrência do dano. Na verdade, as técnicas de detecção de dano baseadas na resposta dinâmica pertencem aos níveis 1, 2 e 3, deixando que a teoria da mecânica da fratura se ocupe do nível 4.

Entre as técnicas baseadas em sinais encontra-se o método *Novelty Detection* (WORDEN; SOHN; FARRAR, 2002), cujo principal objetivo é extrair características dos dados dinâmicos, através de análises estatísticas de eventos extremos para determinar o estado da estrutura. Esta abordagem permite só o nível de detecção um (decidir se tem, ou não, ocorrido dano dentro do sistema). Os outros métodos baseados em sinais (a emissão acústica (CARLOS, 2003), os raios X, as correntes de Eddy (DOHERTY, 1987), os métodos de campo magnético, os métodos foto-térmicos, de ultrassom e a inspeção visual (GOCH et.al., 1999), precisam que a vizinhança do dano seja conhecida *a priori* e que a porção da estrutura a inspecionar-se seja acessível. Além destas limitações, estes métodos só podem detectar dano em/ou próximo da superfície da estrutura e funcionam relativamente bem em estruturas de pequeno porte. As principais vantagens destas técnicas são que evitam os erros de modelagem e os custos computacionais envolvidos nas simulações numéricas. Porém, são ineficientes quando se trata de grandes sistemas estruturais, onde as técnicas fundamentadas na resposta vibratória da estrutura e baseadas em modelos, objeto deste trabalho, são mais promissoras devido a seu caráter global.

Por outro lado, os métodos baseados em modelos permitem estimar a posição do dano e sua severidade mediante o uso do modelo (analítico ou numérico) da estrutura e empregando tanto dados experimentais de vibração

quanto de deslocamentos estáticos (WORDEN; DULIEU-BARTON, 2004; TEUGHELIS et.al., 2002; FILHO et.al., 2000; DUFFEY et.al.,2000; SAMPAIO; MAIA; SILVA, 1999; HJELMSTAD; SHIN, 1996; FRISWELL; MOTTERSHEAD, 1995; CAWLEY; ADAMS, 1979; BAKHTIARI-NEJAD; RAHAI; ESFANDIARI, 2005; SANAYEI; SCAMOLPI, 1991; BANAN; BANAN; HJELMSTAD, 1994). As principais desvantagens dos métodos de detecção baseados na resposta estática são duas: primeiro, existe menos informação a ser usada nos métodos de detecção estáticos, quando comparados com os métodos dinâmicos e, segundo, o efeito do dano pode ser anulado devido a limitações dos caminhos de carga.

1.3 Detecção de dano como Problema Inverso

O problema de detecção de dano baseado em modelos pode ser definido como um problema inverso, isto é, a partir de informação experimental da estrutura (deslocamentos, acelerações, tensões, etc.) busca-se identificar os parâmetros físicos do sistema (módulo de elasticidade, fatores de amortecimento).

Devido a que os dados experimentais geralmente são limitados, podem encontrar-se múltiplas soluções que satisfaçam à formulação do problema inverso. Para contornar esta dificuldade, técnicas heurísticas como as Redes Neurais (RNs) (LEE et. al., 2005; PIERCE; WORDEN; MANSON, 2006; HAMAMOTO; SOMA, 2003; STASZEWKY, 2002; ZANG; IMREGUN, 2001) e os Algoritmos Genéticos (AGs) (SANG-YOUL; SHI-CHANG, 2005; MOSLEM; NAFASPOUR, 2002; RAO; SRINIVAS; MURTHY, 2004), dentre outras, estão sendo utilizadas com sucesso para resolver o problema DD. A idéia central

destas técnicas é substituir o problema inverso de detecção de dano por um conjunto de problemas diretos, geralmente, problemas de otimização, cuja solução permite obter uma representação (vetor) do sistema, onde as variações nos parâmetros da estrutura devidos ao dano são realçadas, permitindo a identificação de possíveis falhas (Figura 1). Devido à presença de múltiplos pontos ótimos no problema DD baseado em modelos, as técnicas de otimização clássicas que empregam gradientes correm o sério risco de falhar na identificação do dano.



Figura 1. Solução do problema de detecção de dano com insuficiente informação mediante formulação de um problema de otimização.

Uma das principais dificuldades encontradas na detecção de dano baseada em modelos é a presença de erros, tanto nas medidas experimentais (ruído) como nos modelos utilizados (de modelagem, na ordem do modelo e nos parâmetros do modelo). O efeito destes erros pode ser incorporado incluindo, por exemplo, ruído nos dados de entrada utilizados durante o

processo de treinamento da rede neural ou nos dados que definem a função objetivo utilizada nos AGs. Já os erros no modelo de elementos finitos geralmente podem ser reduzidos mediante o emprego de técnicas de ajuste de modelos (KWONG; LIN, 2005).

Entre os métodos heurísticos, o *Simulated Annealing* SA (fundamentado nos princípios da mecânica estatística), e o *Particle Swarm Optimization* PSO (baseado num modelo de interação social simplificado), se apresentam como opções promissoras para solucionar o problema DD, como mostrado neste trabalho. Suas vantagens e desvantagens são apresentadas, e uma estratégia para melhorar o desempenho do PSO, através da formulação de um novo algoritmo híbrido, é introduzida.

Finalmente, o uso de modelos numéricos pode ser vantajoso em diversos casos, como por exemplo: permitir um grande número de simulações de pequenas variações no sistema (condições de contorno, condições ambientais, parâmetros físicos, etc.) facilitando o planejamento de ensaios e a colocação ótima de transdutores para realizar medições necessárias para a DD. O modelo também pode ser usado na fase de treinamento de sistemas de detecção de dano baseados em redes neurais, onde a geração de grandes quantidades de dados é requerida.

1.4 Objetivos da Tese

Desenvolver um novo algoritmo heurístico híbrido autoconFigurado, para solucionar o problema DD, empregando modelos de elementos finitos (MEF) que identifique dano em nível de elemento. A técnica busca determinar a posição do dano dentro da estrutura e fazer uma estimativa de

sua severidade (extensão). O método utiliza funções de resposta em frequência e / ou dados modais experimentais correlacionadas com parâmetros de dano definidos no modelo de elementos finitos (MEF) da estrutura. Portanto, as hipóteses básicas da técnica para detecção de dano desenvolvida nesta pesquisa são: que a resposta global da estrutura permanece linear após a ocorrência do dano e que existe um modelo analítico / numérico validado da estrutura intacta.

Realizar uma avaliação rigorosa do algoritmo desenvolvido mediante comparação de seu desempenho com o das técnicas existentes (quando for necessário) empregando casos de estudo simulados. A comparação incluirá os seguintes critérios de avaliação: Precisão, Estabilidade, Robustez e Custo Computacional, na identificação de dano. O algoritmo desenvolvido visa melhorar os anteriores critérios para a solução do problema DD.

Quantificar as vantagens e limitações do algoritmo proposto em termos da extensão mínima de dano que pode ser detectado em presença de ruído.

Programar, utilizar e combinar os algoritmos *Nelder e Mead*, *Particle Swarm Optimization (PSO)* e *Simulated Annealing (SA)* para solucionar o problema inverso de detecção, definido para a DD nesta pesquisa. A combinação tem como objetivo melhorar o desempenho dos métodos.

1.5 Alcance

A metodologia aqui estudada pode ser empregada para avaliar diversos tipos de danos, tais como: problemas com ligações, solda, corrosão, fissuras, perda de elementos, e variações das condições de contorno em sistemas estruturais. O comportamento da estrutura pode ser linear o não linear, e a

única condição para a aplicação do algoritmo apresentado é ter um modelo validado para determinar a resposta da estrutura.

Nos exemplos numéricos apresentados, se supôs que o dano afetou diretamente a rigidez do sistema e que o comportamento, antes e após o dano, é linear.

1.6 Síntese do conteúdo da Tese

O conteúdo desta tese está organizado como descrito a seguir. O capítulo 2, fornece uma revisão da literatura relevante relacionada com esta pesquisa. Basicamente, a revisão contém um resumo dos métodos de detecção de dano via respostas dinâmicas.

O capítulo 3, descreve as bases teóricas da detecção de dano. Introduz idéias básicas como o conceito de problema inverso, os métodos de correlação entre resultados teóricos e experimentais e os modelos de dano empregados.

O capítulo 4, fornece uma revisão completa dos algoritmos PSO, SA e AG. Estas técnicas heurísticas apresentam características que as tornam atraentes para solucionar o problema de detecção de dano. Os mecanismos de cada algoritmo são explicados e o funcionamento dos operadores presentes em cada um deles é descrito.

No capítulo 5, se apresenta o novo algoritmo híbrido PSOS. O PSOS é uma combinação do PSO e do método Simplex. Sua principal característica é sua capacidade de autoconfiguração. Adicionalmente, ele apresenta boa estabilidade e precisão, o qual o torna uma ferramenta útil para a detecção de dano.

No capítulo 6, o potencial e as limitações do PSOS são analisados. O desempenho do PSOS é testado em três funções teste e seu desempenho é comparado com o dos algoritmos descritos no capítulo 5.

No capítulo 7, se analisam dois casos difíceis de detecção de dano usando dados modais. Uma comparação entre os resultados do PSOS e o SA na identificação do dano numa viga em balanço é apresentada.

O capítulo 8, oferece três exemplos de detecção de dano usando FRFs e o PSOS. Aqui, o potencial e as limitações do novo algoritmo são analisadas.

Finalmente, no capítulo 9, se apresentam as conclusões e contribuições desta tese, junto com algumas idéias para trabalhos futuros.

2 Detecção de dano-Estado da arte

2.1 Introdução

Neste capítulo, apresenta-se uma ampla revisão crítica dos métodos para detecção de dano em estruturas com a finalidade de contextualizar o trabalho desenvolvido nesta tese. A revisão inclui os trabalhos mais relevantes na área relacionados com este estudo.

Teoricamente, mudanças no comportamento vibratório da estrutura refletem alterações nos parâmetros físicos e nas condições de serviço e, por esta razão, podem ser utilizadas como indicadoras de dano. Os primeiros trabalhos que estudam o comportamento dinâmico de barras com descontinuidades de rigidez datam da década de 40 do século passado (THOMPSON, 1943; KIRMSHER, 1944). Porém, se passam quase 25 anos até aparecer o primeiro artigo a propor especificamente um método de detecção de dano via medidas de vibração (LIFSHITZ; ROTEM, 1969). Nesse trabalho, de natureza experimental, os autores correlacionam variações do módulo dinâmico, medido da curva tensão-deformação, sob carregamento dinâmico, com o dano introduzido em corpos de prova compostos de elastômeros.

Na década de 70, a pesquisa na área ganhou grande impulso devido à constatação de que a simples inspeção visual de estruturas (pontes, aeronaves), não era suficiente para determinar sua confiabilidade (LIU; YAO, 1978). Durante os últimos trinta anos, a literatura especializada neste campo tem se multiplicado, mostrando a vitalidade e fôlego desta área de pesquisa. São exemplos, as excelentes revisões feitas por Doebling et.al. (1996, 1998), Salawu (1997), Fanning e Carden (2004), e a do FIB Europe Task Group 5-1-SHM Guidelines (2002); assim como o surgimento, em 2002, da revista STRUCTURAL HEALTH MONITORING dedicada especificamente à publicação de pesquisas teóricas e experimentais na área.

Diversos congressos científicos internacionais sobre temas como estruturas adaptativas, monitoração da saúde estrutural, detecção de dano, materiais e sistemas inteligentes, redes de sensores e isolamento de vibrações, vem sendo realizados cada vez com maior frequência e fornecem um espaço para intercâmbio de idéias entre engenheiros, arquitetos e cientistas que trabalham no campo da detecção de dano. Entre os eventos recentes mais importantes se podem mencionar os seguintes: The first International conference on Structural Health Monitoring of Intelligent Infrastructure (SHMII-1'2003), The Second International conference on Structural Health Monitoring of Intelligent Infrastructure (SHMII-2'2005), The 5th Conference on Damage Assessment of Structures (DAMAS 2003), The 6th Conference on Damage Assessment of Structures (DAMAS 2005), The XXIII International Modal Analysis Conference (IMAC XXIII-2005), The 2nd International Workshop on Structural Health Monitoring of Innovative

Civil Engineering Structures (SHM ISIS 2004) e The Fourth World Conference on Structural Control and Monitoring (4WCSCM 2006).

Nas seguintes seções, se apresenta uma classificação unificada dos métodos de detecção de acordo com sua dependência (ou não) de um modelo estrutural.

2.2 Técnicas Baseadas em Modelos (TBM)

Em situações práticas é quase impossível dispor de dados experimentais suficientes para caracterizar todos os possíveis estados (e ambientes) nos quais a estrutura estará, e, mesmo o anterior sendo possível, o custo desses testes, em larga escala, inviabilizaria sua realização. A anterior é talvez a principal razão pela qual os modelos numéricos são hoje amplamente usados em engenharia, já que, eles permitem estudar, com um custo relativamente baixo, fenômenos de interesse que não podem ser medidos diretamente. Este argumento pode ser reforçado pela crescente eficiência dos processadores, pela disponibilidade de memória e de técnicas de computação paralela.

Atualmente, os métodos numéricos permitem simular, passo a passo, o comportamento de grandes sistemas estruturais, o que possibilita a previsão, a comparação e a validação de resultados do modelo e, portanto, seu uso na solução do problema DD.

Nos últimos anos tem surgido uma nova área de pesquisa conhecida como Monitoramento da Saúde Estrutural MSE (SHM, em inglês), que busca a integração de modelos refinados de estruturas com sensores inteligentes, colocados em lugares estratégicos, e procedimentos inovadores de

processamento e aquisição de sinais, com o objetivo final de monitorar em tempo real, o desempenho da estrutura e fornecer uma avaliação confiável sobre sua segurança. É dentro de este ambiente que o problema de detecção pode ser solucionado.

As TBMs, desenvolvidas a partir da década de oitenta (século XX), realizam basicamente uma comparação das respostas experimentais com os resultados obtidos da análise do modelo da estrutura (para análises estruturais o modelo mais utilizado é o Modelo de Elemento Finitos - MEF), com a finalidade de determinar a presença de dano.

Os dados medidos durante um experimento dinâmico podem ser de três tipos (FRISWELL; PENNY, 1997): dados no domínio do tempo, dados no domínio da frequência e dados modais. As excitações e respostas no domínio do tempo (séries temporais) podem ser obtidas a partir de acelerômetros ou de sensores piezelétricos e geralmente são de caráter digital. As funções de resposta em frequência (FRFs) são obtidas no caso em que se conhece tanto a excitação como a resposta, a partir das séries temporais mediante o algoritmo da Transformada Rápida de Fourier (FFT). As FRFs, em sistemas lineares, servem de base para o cálculo dos parâmetros modais (frequências naturais, amortecimento modal e formas modais), através de métodos de ajuste, que essencialmente, acertam os coeficientes de uma expressão teórica para a FRF, de modo tal, que o modelo teórico reproduza, de melhor forma possível, os dados experimentais. Para sistemas não lineares é possível obter FRFs de ordem superior (HFRFs) que podem ser usadas também para detecção de dano (STORER; TOMLINSON, 1993).

A comparação entre resultados numéricos e dados experimentais, obtidos, por exemplo, da análise modal, deve levar em conta, entre outros fatores: a presença de amortecimento na estrutura real (difícil de modelar no MEF), os problemas causados pelas diferenças entre o número de graus de liberdade do MEF e o número de pontos onde é medida a resposta na estrutura e à presença tanto de erros de modelagem como de erros nas medições (inevitáveis, porém, mensuráveis). Para contornar alguns destes problemas, existem dois caminhos, descritos a seguir. O primeiro é o emprego de técnicas de redução / expansão de modelos, como os propostos por Berman e Nagy (1983), Guyan (1965) e Kim e Bartkowics (1993). No segundo, podem ser empregadas as FRFs (ou HFRFs) ou um conjunto de séries temporais junto com o modelo numérico diretamente no esquema de detecção (DASCOTTE; STROBBE, 1998; FANNING; CARDEN, 2003; WORDEN; TOMLINSON, 2001). A primeira opção, o uso de HFRFs, possibilita o estudo de sistemas não lineares.

As principais vantagens da abordagem TBM são sua capacidade para identificar danos a partir da resposta dinâmica global da estrutura, até o nível três definido por Rytter (1993) e sua versatilidade, já que, podem ser aplicadas tanto para sistemas lineares como para sistemas não lineares.

Na maioria dos casos de interesse prático, os modelos estruturais tomam a forma do MEF. O MEF tem sido usado de maneira ampla em aplicações industriais e de pesquisa devido a que pode produzir uma boa representação da estrutura real. Porém, a predição do MEF não sempre é exata. Erros e imprecisões podem surgir devido a fatores como: estimativas imprecisas das propriedades dos materiais; modelagem pobre das condições

de contorno; qualidade baixa da malha de elementos empregada; omissão do amortecimento ou uso de um modelo de amortecimento inapropriado. De forma geral, como mencionado anteriormente, estes problemas podem ser tratados mediante as técnicas de ajuste de modelos.

Por outro lado, as TBMs podem ser divididas nos seguintes três grupos dependendo do tipo de dados experimentais utilizados. No primeiro, têm-se os métodos modais, no segundo, os métodos baseados no domínio da frequência e no terceiro, os métodos baseados no domínio do tempo. A característica principal das TBMs contempladas dentro destes três grupos é que empregam técnicas convencionais de modelagem e procedimentos clássicos de análise numérico para resolver o problema DD.

Nas seções seguintes será feita uma análise dos métodos pertencentes aos grupos acima definidos.

2.2.1 Métodos que empregam dados Modais

Este grupo de métodos de detecção está estreitamente ligado à área conhecida como análise modal experimental (ou identificação paramétrica de sistemas) e tem se tornado, desde os anos 1970, numa ferramenta muito popular devido a seu amplo espectro de aplicações, que vai desde problemas de acústica até detecção de dano.

A análise modal é uma das estratégias fundamentais para abordar problemas de dinâmica de estruturas. Apesar de seu uso estendido, é basicamente uma teoria linear, o qual pode ser uma séria restrição no mundo real, onde o comportamento não linear tem um reconhecido efeito sobre a resposta dinâmica das estruturas. A maioria dos métodos de

detecção apresentados na literatura recente (FOX, 1992; FRISWELL; MOTTERSHEAD, 1995; JAUREGUI; FARRAR, 1996a-b; LU; REN; ZHAO, 2002; KIM, et.al., 2003; PANDEY; BISWAS; SAMMAN, 1991; PANDEY; BISWAS, 1994-1995; SALAWU; WILLIAMS, 1994; WAHAB; DE ROECK, 1999), supõe a invariância da freqüência naturais, dos fatores de amortecimento, das formas modais e das FRFs com relação à força de excitação durante um ensaio real. Na verdade, para um sistema não linear, todas essas quantidades são funções do nível de excitação. A razão fundamental da popularidade da informação modal para detecção de dano reside em que a conversão de séries temporais para dados modais reduz significativamente a quantidade de dados a serem analisados, sem perda apreciável de informação para uma dada banda de freqüências.

Uma pesquisa pioneira na área, empregando informação modal de um sistema linear e a técnica de ajuste de modelos, foi apresentada por Baruch (1978). A formulação do método se baseia na minimização da norma euclidiana da diferença entre a matriz de rigidez analítica da estrutura $[K]$ e uma matriz de rigidez ajustada $[K_a]$, que, além de satisfazer a equação dinâmica de equilíbrio e ser simétrica, deve ser capaz de reproduzir os auto valores e auto vetores do sistema original. O autor utiliza o método dos multiplicadores de Lagrange para calcular uma solução explícita da matriz $[K_a]$. A matriz de massa do modelo analítico é suposta como correta. O principal problema do método é que a nova matriz $[K_a]$ sofre alterações radicais (como por exemplo, perda da esparcidade) o que impede seu uso direto para identificação de dano

Um dos primeiros trabalhos neste campo é o apresentado por Cawley e Adams (1979). Nele, os autores desenvolvem e aplicam um método baseado na correlação entre um MEF (sem incluir amortecimento) de uma placa de alumínio e as variações de frequência para detectar defeitos (problema 2D). A região danificada da placa é localizada dentro das restrições impostas pela simetria. A presença de dano é indicada de forma pouco precisa a partir da variação nas frequências naturais.

Lieven e Ewins (1988) propõem o COMAC (*Coordinate Modal Assurance Criterion*) como método de comparação e correlação entre formas modais. O COMAC correlaciona dois conjuntos de formas modais, que podem ser obtidos, tanto de forma experimental quanto de forma analítica, e identifica as coordenadas responsáveis pela falta de correlação. A técnica é utilizada para detectar regiões inconsistentes (por exemplo, com aumentos de massa) dentro de uma estrutura de molas e massa com oito graus de liberdade.

Stubbs, Kim e Toppole (1992), apresentam um método baseado no decremento na energia de deformação modal entre dois graus de liberdade, definido pela curvatura das formas modais. Neste trabalho, é estabelecido um índice de dano normalizado para cada elemento que permite definir a posição do dano na estrutura. A principal desvantagem do método é a dificuldade para calcular as derivadas e integrais necessárias para determinar o índice, quando as formas modais são estimadas a partir de poucos pontos de medição. Este problema está presente em todos os métodos que utilizam variações da energia modal.

Zimmerman e Simmermacher (1995) fazem uma extensão da teoria da Mínima Perturbação do Rank (MRPT, em inglês) para o caso em que existem

múltiplos resultados da análise modal (ou de análises estáticas) disponíveis. A teoria MRPT determina uma matriz de rigidez de perturbação simétrica que melhora a correlação entre os dados medidos e os dados analíticos, possibilitando a identificação de danos dentro da estrutura.

Cornwell, Doebling e Farrar (1999) estudam métodos baseados nas mudanças da energia de deformação para localizar danos. Originalmente, estes métodos foram empregados para vigas (curvatura unidimensional). O estudo apresentado é uma generalização do método de Stubbs, Kim e Toppole (1992) para estruturas 2D.

Filho, Roitman e Magluta (2000), utilizam a metodologia de ajuste de modelos por matriz ótima, propondo uma solução alternativa ao problema da esparcidade real dos modelos, que na verdade é uma modificação do método de Kabe (1985), e, mediante simulações numéricas com diferentes cenários de dano em pórticos, conseguem indicar, em quase todas as simulações, a região do dano para cenários de dano livre de ruídos. Para medições (frequências e formas modais) com ruídos simulados, observou-se uma severa queda na sensibilidade do método quando usados mais de três modos na identificação.

Teughels, Maeck e De Roeck (2002) propõem um método de detecção de dano mediante ajuste do modelo de elementos finitos, usando funções de dano para representar a distribuição de rigidez do sistema. A função objetivo utilizada é formulada em termos dos vetores residuais de frequência e formas modais. Através do método de otimização TRN *standard -Trust Region Newton method-* (NOCEDAL; WRIGTH, 1999), eles conseguem detectar, localizar e quantificar o dano em uma viga de concreto reforçado. A

abordagem apresentada pelos autores visa reduzir o número de parâmetros a serem ajustados, evitando o ajuste individual da rigidez de cada elemento, resultando num método robusto de otimização e gerando uma distribuição suave das propriedades do modelo, capazes de representar as regiões danificadas dentro da estrutura.

No trabalho de Kim e Stubbs (2003) é apresentada uma metodologia que relaciona as variações em energia modal com as mudanças nas frequências naturais, ocasionados pela presença de fissuras. Em teoria, o método proposto pode detectar a presença de fissuras de diversos tamanhos. Na prática a identificação pode ser difícil devido a incertezas nas medições causadas por efeitos ambientais.

Na maioria dos métodos mencionados nesta seção, supõe-se que a matriz de massa $[M]$ não muda. Isto significa que a base de referência da técnica proposta é a matriz $[M]$. Esta suposição tem um significado profundo, já que como demonstrado por Baruch (1997), as matrizes da massa e rigidez não podem ser identificadas simultaneamente empregando unicamente informação modal, devido a que existe infinito número de soluções para este problema.

2.2.2 Métodos que Empregam Dados no Domínio da Frequência

As vantagens de se usar FRFs em vez de dados modais são várias. As FRFs podem ser medidas diretamente na estrutura sem necessidade de nenhum passo intermediário e adicionalmente, as FRFs fornecem informação sobre uma banda de frequência e não somente para frequências

específicas, como é o caso da informação modal. Finalmente, quando as FRFs são usadas para definir a função objetivo, nos métodos de detecção de dano, não é preciso medir uma coluna ou uma fila inteira da matriz de funções de resposta, abaixando-se os custos de realizar experimentos extensos.

Diversos autores tem usado FRFs para identificação de sistemas e detecção de dano. Um método que utiliza a curvatura das FRFs foi proposto por Sampaio, Maia e Silva (1999). Esta técnica é uma extensão do método proposto por Pandey e Biswas (1991), para todas as frequências medidas e não só para as frequências e formas modais. O método é testado em quatro cenários de dano, numa ponte real (vão total de 13.3 m), mostrando-se eficiente na identificação da posição da falha para o cenário mais severo e completamente ineficaz, nos outros três casos. O mérito principal do método é sua simplicidade, já que, não precisa de análise modal para sua aplicação.

Jones e Turcote (2002) empregam anti-ressonâncias para identificar dano em uma treliça e mostram que seu uso pode ser um substituto adequado dos dados modais.

Fanning e Carden (2003) introduzem um método de detecção baseado na medida de uma única FRF (técnica *Single input – Single output* (SiSo)) junto com um modelo simplificado para determinar a FRF de receptância, nos casos em que a estrutura sofre uma variação de rigidez num número limitado de elementos. Em geral, diferentes combinações de dano podem produzir a mesma diferença entre a FRF da estrutura sadia e a FRF danificada. Para contornar este problema, os autores identificam as possíveis combinações de dano (para cada frequência de ressonância) que

produzem a mesma mudança na FRF medida e comparam esses conjuntos para determinar o evento de dano comum a todos. Este dano comum será o dano real na estrutura. O método só pode ser aplicado a estruturas com comportamento linear e não deixa claro como poderia ser aplicado em casos onde mais de dois locais danificados existem.

Zimmerman, Simmermacher e Kaouk (2005), empregam a teoria da Mínima Perturbação do Rank (MRPT) junto com FRFs para detecção de dano. Este método pode ser classificado como uma técnica de ajuste de modelos. O trabalho estende pesquisas prévias dos autores, onde eram utilizados dados modais e dados estáticos. A vantagem de usar FRFs diretamente é que elas oferecem a possibilidade de reduzir o tempo de experimentação e de análise. Por outro lado, o uso de FRFs no esquema para DD tem o inconveniente que o amortecimento deve ser incluído no MEF. Esta incorporação é importante para obter uma boa correlação entre as FRFs medidas e as calculadas de forma analítica. Devido à dificuldade de modelar o efeito do amortecimento de forma exata, o amortecimento proporcional viscoso é geralmente usado junto com o MEF, porém, outros modelos de amortecimento podem ser empregados (FRISWELL; MOTHERSHEAD, 1995).

2.2.3 Métodos que Empregam Dados no Domínio do Tempo

Os métodos neste grupo utilizam séries temporais como informações básicas para solucionar o problema DD. Os benefícios desta abordagem são múltiplos: primeiro, à diferença da análise modal, não está restrita ao estudo de sistemas lineares e pode ser usada na análise de sistemas com fontes

arbitrarias de incertezas (por exemplo, físicas, ambientais e de interfase); segundo, evita o processo de compressão de dados, realizado durante a análise modal, que pode causar perda de informação sobre o comportamento dinâmico da estrutura quando existem fissuras ou problemas de ligação; e, por último, podem ser usados tanto para vibração livre como para forçada, com o sem conhecimento da força. Sua principal desvantagem é que podem requerer de um esforço computacional considerável para calcular a resposta temporal da estrutura.

Ostachowicz e Krawczuk (1990) apresentam uma pesquisa sobre identificação de fissuras em vigas a partir da amplitude máxima da resposta temporal. Recentemente, no trabalho de Garcia e Osegueda (2000), o uso dos coeficientes de regressão do Modelo ARMA (*Autoregressive Moving Average*), obtidos de séries temporais, tem sido empregados para identificação de dano. Um outro algoritmo para DD que emprega séries temporais e o modelo ARMA é o proposto por Nair, Kiremidjian e Law (2006). Nesse trabalho, é mostrado que à medida que a rigidez da estrutura diminui, devido ao dano, os coeficientes das três primeiras componentes AR do modelo mudam. Como estes coeficientes, geralmente, contem informação sobre as frequências naturais e sobre os fatores de amortecimento do sistema, sua variação permite a identificação do dano.

2.3 Técnicas Baseadas em Sinais (TBS)

A idéia destes métodos é empregar unicamente dados experimentais para solucionar o problema de detecção, dispensando assim o uso de modelos. Neste grupo, além das técnicas convencionais (não destrutivas)

para detecção de dano (Emissão Acústica, Corrente de Eddy) podem mencionar-se os métodos conhecidos como Detecção de Anomalias ou Detecção Eventos Extremos (*Novelty detection* ou *Outliert analysis*). O propósito desta técnica é estabelecer um conjunto de características que definam a condição normal da estrutura e utiliza-las como referência, estabelecendo um padrão de operação normal do sistema. Durante etapas posteriores de monitoramento, as novas características estimadas são comparadas com a referência, e, se qualquer desvio significativo aparecer, o algoritmo indicará a presença da anomalia. Como anotado por Worden e Dulieu-Barton (2004), os métodos de detecção de anomalias estão limitados ao nível um definido por Ritter (1993).

No trabalho apresentado por Sohn e Farrar (2001), a detecção do dano em um sistema de massas e molas com 8 graus liberdade é realizada analisando unicamente dados de aceleração do sistema estudado. Zhang (2006) reporta uma técnica para detectar fissuras de fadiga baseada no uso de sensores piezelétricos embebidos em películas de tinta, porém, uma disposição geométrica complexa desses sensores é necessária, em alguns casos, a fim de melhorar a sua resolução espacial. Recentemente, Leong et. al. (2005), empregaram vibrômetros a laser para medir a velocidade de propagação de ondas ultrasônicas (ondas de LAMB) numa placa de alumino, isto permitiu detectar uma fissura de fadiga localizada no centro da placa. Esta é uma área de pesquisa promissora, porém, são necessários avanços nos processo de análise de sinais para incrementar a sensibilidade do método.

Devido a que a análise modal é realizada no domínio da frequência, a maioria dos métodos de detecção de dano emprega a Transformada Rápida de Fourier (FFT). Neste contexto, a Transformada Wavelet (TW) pode ser vista como uma extensão de Transformada de Fourier, com a vantagem de fornecer boa resolução, tanto no tempo como na frequência, para um dado sinal (TAHA et.al., 2006). Por esta razão, a TW tem ganho muita popularidade nos últimos tempos como ferramenta para processamento de sinais e vem sendo aplicada ao problema de detecção de dano. Uma excelente revisão sobre aplicação do método da TW na área de máquinas rotatórias é apresentada por Peng e Chu (2004). Qi et.al. (1997), mostraram que mediante aplicação da TW ao análise de sinais de emissão acústica (EA), o dano pode ser detectado e quantificado. A TW foi usada por Rus et.al. (2004), para diferenciar entre o estado degradado e o intacto em materiais compostos.

2.4 Métodos Heurísticos

Neste grupo está incluída uma classe emergente de métodos para solucionar o problema DD. Sua característica principal é combinar técnicas clássicas de modelagem com novos paradigmas de computação, como as redes neurais, os algoritmos genéticos e teoria de conjuntos difusos, entre outros. O tipo de informação experimental usado pode ser qualquer um dos mencionados anteriormente ou uma combinação de eles. Os métodos clássicos, apesar de seu extenso uso, sofrem ainda algumas limitações como problemas de divergência e instabilidade nos cálculos numéricos e são

susceptíveis de ficar presos em pontos ótimos locais (S-Y LEE; S-H WOOH, 2005).

Recentemente, algoritmos heurísticos de otimização vem sendo empregados com sucesso para procurar soluções factíveis para o problema de detecção de dano, onde a complexidade do problema e/ou o tempo disponível para sua solução não permitem uma solução exata.

Mares e Surace (1996), apresentam um dos primeiros métodos de detecção de dano via resposta dinâmica da estrutura, baseados em Algoritmos Genéticos (AGs) e na teoria da análise modal convencional. Os AGs são um procedimento de busca de pontos ótimos globais, que imita os processos da seleção natural e da evolução descritos por Charles Darwin em seus famosos trabalhos do século XIX. Mediante a seleção de parâmetros num modelo de elementos finitos da estrutura (viga), e utilizando AGs, os autores minimizam a diferença entre as propriedades dinâmicas calculadas e as medidas. Os parâmetros obtidos após minimizar o vetor de força residual modificada, identificam a posição e severidade do dano. Além da capacidade de identificação, o método proposto exhibe robustez nas simulações numéricas notadamente em relação à influencia de ruídos nos dados experimentais.

Moslem e Nafaspour (2002), propõem um procedimento de duas etapas, baseado em dados modais experimentais incompletos, no MEF da estrutura e no AG, para determinar a posição e severidade do dano em treliças 2D. Na primeira etapa, mediante o emprego do método da força residual, são identificadas as possíveis áreas afetadas. Na segunda etapa, o AG é utilizado para determinar a extensão do dano nas áreas identificadas

no passo anterior. O algoritmo proposto mostrou grande sensibilidade com relação ao método de redução de modelo empregado. Os métodos mais refinados de redução (condensação), como o SEREP -*System Equivalent Reduction Expansion Process*- e o IRS -*Improved Reduced System*- (O'CALLAHAN et.al., 1989), produziram os piores resultados quando comparados com o método de Guyan, (1965). Au et. al. (2003), utilizam um algoritmo micro-genético (população pequena) para detectar dano em vigas e avaliar a influência do ruído e de medições incompletas na identificação. Pawar e Ganduli (2003), empregam os AGs para projetar automaticamente um sistema *fuzzy* que maximiza a precisão do sistema de detecção de dano. Bland e Kapania (2004) estudaram a efetividade de um AG híbrido proposto por eles, na identificação de dano estrutural em placas enrijecidas. Os resultados mostram que, é possível solucionar o problema de detecção, de forma eficiente, combinando os atributos positivos dos AGs com as técnicas de programação não linear convencionais. Nesse estudo, os AGs são empregados para obter uma estimativa dos melhores pontos para iniciar uma busca mais refinada, fundamentada em algoritmos de otimização baseados em sensibilidade (derivadas).

Uma desvantagem dos AGs (binários) é que eles podem consumir muito tempo de computação no processo de codificar-decodificar e durante esta ação, alguns genes importantes podem ser perdidos acarretando perda de precisão na solução procurada. Para evitar este problema, tem sido desenvolvidos AGs que empregam uma representação real das variáveis. Um outro problema que enfrentam os AGs é a convergência prematura, a qual, por óbvias razões, deve ser evitada.

Recentemente, tem sido realizadas diversas tentativas para melhorar a capacidade dos AGs de fugir de pontos ótimos locais e impedir sua convergência nas etapas iniciais, constituindo uma área aberta e muito ativa de pesquisa (HAUPT; HAUPT, 1998, WONG; HAMOUDA, 2000).

Criadas por pesquisadores das ciências da computação, as Redes Neurais Artificiais (RNAs) vem sendo empregadas recentemente como uma ferramenta para solucionar o problema DD. Basicamente, as RNAs podem ser consideradas como aproximadores universais de funções. Estudos recentes mostram que uma rede neural de três camadas pode ser usada com sucesso no problema de detecção de dano (LUO; HANAGUD, 1997; ZENG, 1998). Leath e Zimmerman (1992), empregam uma rede neural conhecida como *Multilayer Backpropagation* (BP) para identificar dano numa viga em balanço modelada com quatro elementos finitos, utilizando as duas primeiras frequências naturais. Os autores desenvolvem um algoritmo de treinamento, que pode criar uma RNA que ajusta os dados, usando um número mínimo de neurônios. O dano é modelado como uma redução no módulo de elasticidade de até 95%. O método identifica a posição do dano com um erro máximo de 34%.

Tsou e shen (1994), usam uma RNA-BP para identificar dano no oscilador de 8 graus de liberdade proposto por Kabe (1985). O dano é introduzido diminuindo o valor da rigidez das molas. Os autores testam diversas configurações para a rede neural (40, 60 e 100 nós ocultos). Na fase de treinamento, utilizam 105 exemplos para fazer o mapeamento do vetor de força residual para a variação de rigidez. O método identifica a severidade do dano dentro de 5% de exatidão. O modelo de dano estudado é linear.

Yun e Bahng (2000), utilizam uma RNA-BP para identificar dano, empregando frequências naturais e formas modais como entradas para a RNA. A detecção é realizada dividindo o sistema em subestruturas para reduzir o número de incógnitas.

Lee et al. (2005), utilizam a diferença entre os componentes das formas modais antes e após acontecer o dano, como dados de entrada para a rede neural (estes dados são menos sensíveis que a formas modais a erros de modelagem). Os dados de saída contem índices de dano para cada elemento do MEF empregado para gerar os dados. Para a ponte estudada pelos autores, localizada na Coreia, a posição do dano foi estimada com sucesso em todos os casos analisados, porém, sua severidade teve falsos positivos em diversas posições.

Como sabido, as RNAs fornecem uma metodologia geral de mapeamento entre um conjunto de dados de entrada e um conjunto de dados de saída. A rede é treinada pra minimizar de forma recursiva uma função de erro entre a resposta objetivo e sua resposta atual, possibilitando desta forma a identificação do dano. De forma geral, a fase de treinamento ou aprendizado é um dos pontos críticos da detecção de dano usando RNAs. Nesta etapa, pode surgir o problema de sobre-treinamento (PIERCE; WORDEN; MANSON, 2006), fazendo com que a rede seja afetada pela presença de ruído. Adicionalmente, para realizar o treinamento, é necessário um grande número de medições experimentais, o qual pode ser um sério problema em estruturas civis, devido à dificuldade de sua obtenção.

3 Detecção de Dano: Bases Teóricas

3.1 Introdução

A identificação de dano a partir de respostas dinâmicas se fundamenta no fato de que variações locais na rigidez, no amortecimento ou nas condições de contorno afetam as características dinâmicas globais de estrutura. O problema de dinâmica de estruturas pode ser conceitualmente expresso da seguinte maneira:

$$D = T(F, M, C, P)F \quad (1)$$

onde, D é um vetor que contém as respostas (deslocamentos, acelerações e velocidades) do sistema; T é uma matriz de transformação; F é um vetor de forças; M é um vetor que contém as propriedades dos materiais; C é um vetor que caracteriza as condições de contorno; P é um vetor de coeficientes relacionados com o modelo. De acordo com o tipo de informação procurado na solução da Equação (1), os problemas de dinâmica de estruturas podem ser tratados como problemas diretos (PD) ou problemas inversos (PI). O primeiro grupo consiste em calcular o vetor D , dados o vetor F e a matriz de

transformação T para um conjunto de vetores F , M , C e P conhecidos. O segundo (PI), pode, por sua vez, ser subdividido em: a) determinar o vetor F (causas) a partir das respostas medidas (efeitos) e a matriz de transformação T (conhecida) e este problema é chamado de problema de identificação de causas; b) identificar os parâmetros nos vetores M , C e P , a partir dos vetores D e F conhecidos. Este tipo de problema é denominado problema de identificação paramétrica (PIP) e é abordado nesta pesquisa.

Os PDs têm sido amplamente estudados na área da mecânica e da engenharia de estruturas (propagação de ondas, cálculo de deslocamentos em estruturas, etc.). De forma geral, uma vez estabelecidas as causas (vetor F) e os vetores M , C e P , o problema pode ser solucionado de forma analítica ou numérica (por exemplo, via MEF, ou via o método dos elemento de contorno MEC).

Por outro lado, os PIs na área de engenharia têm recebido muita atenção nos últimos anos (TARANTOLA, 2004; GLADWELL, 1997). Estes problemas incluem a detecção de dano em estruturas, a identificação das propriedades dos materiais a partir de respostas estáticas e dinâmicas, a determinação de forças, o ajuste de modelos, o reconhecimento de voz e a proteção contra a corrosão, dentre outros.

Na literatura recente, o problema inverso de obter um modelo de segunda ordem (representado pelas matrizes de massa, rigidez e amortecimento), a partir de um modelo de primeira ordem, identificado no espaço de estados, vem sendo estudado por diversos autores (LUS, et. al., 2003; ALVIN; PARK, 1994). A principal dificuldade que esta abordagem enfrenta se dá nos casos em que o número de graus de liberdade do sistema

é maior que o número de modos identificados, neste evento, as matrizes físicas, de ordem reduzida, que podem ser obtidas, não necessariamente permitem interpretações físicas e por tanto não fornecem uma indicação direta da presença de dano. Formulações alternativas para solucionar este problema constituem uma área de pesquisa aberta.

No campo da detecção de dano em estruturas civis, os vetores F , M , C e P são conhecidos de forma parcial, devido a que o número de sensores usados é, quase sempre, menor que os graus de liberdade do modelo. Por este motivo, a informação restante é geralmente obtida de forma artificial com base em idealizações (por exemplo, empregando técnicas de expansão modal ou de redução de modelos). Obviamente esta abordagem é imprópria, devido a que, matematicamente, a DD é um problema mal posto, isto é, pequenos erros nos dados de entrada podem produzir grandes erros na resposta. Para tentar contornar este problema, uma possível solução é tratá-lo sem hipóteses artificiais sobre os valores de F , M , C e P e supor que existe uma solução direta do problema, via o MEF (que é um modelo de segunda ordem), por exemplo. Neste contexto, o PI com insuficiente informação pode ser solucionado mediante a formulação de um problema de Programação Não Linear (PNL). Desta forma, é possível substituir a solução do PI por uma série de soluções do PD. A grande vantagem desta abordagem é que a solução direta (ou análise direta) pode, na grande maioria dos casos, ser obtida.

3.2 Correlação entre Resultados Teóricos e Experimentais

Quando dados experimentais de vibração da estrutura estão disponíveis, as medidas de correlação entre os dados analíticos e os medidos fornecem informação útil sobre a qualidade do modelo. Esta correlação pode ser usada para ajustar o modelo analítico para o problema de detecção de dano. Uma medida de correlação muito popular, para dados modais é o *Coordinate Modal Assurance Crtiterion* (COMAC), proposto por Lieven e Ewins (1988).

$$(COMAC(i)) = \frac{\sum_{p=1}^{P \max} |\phi_i^{a,p} \cdot \phi_i^{m,p}|^2}{\sum_{p=1}^{P \max} (\phi_i^{a,p})^2 \cdot \sum_{p=1}^{P \max} (\phi_i^{m,p})^2} \quad (2)$$

onde, i representa o grau de liberdade analisado; $Pmax$ o número total de pares de modos correlacionados; $(\phi_i^{a,p})$ é o i -ésimo elemento do modo analítico estudado, $(\phi_i^{m,p})$ é o i -ésimo elemento do modo experimental. O resultado do COMAC varia entre zero e um. Um valor de COMAC igual a um indica correlação perfeita. O análogo do COMAC no domínio da frequência é chamado de FRAC (*Frequency Response Assurance Criterion*) e foi proposto por Heylen e Avitabile (1998). O FRAC para o i -ésimo grau de liberdade é dado por:

$$FRAC_i = \frac{|H_x^c(\omega_j)_i H_a(\omega_j)_i|}{|H_x^c(\omega_j)_i H_x(\omega_j)_i| |H_a^c(\omega_j)_i H_a(\omega_j)_i|} \quad (3)$$

onde, $H_a(\omega_j)_i$ é a FRF analítica ou numérica, calculada no i -ésimo grau de liberdade nas frequências sucessivas ω_j , $H_x(\omega_j)_i$ é o vetor de FRFs

correspondente medido experimentalmente, e a letra c representa a transposta complexa conjugada. Esta medida varia entre zero e um, novamente, um indica uma correlação perfeita e zero indica uma correlação nula. O FRAC contém informação para um par de pontos específicos.

Existem outras medidas de correlação para dados no domínio da frequência como o *Global Shape Criterion* (GSC) e o *Global Amplitude Criterion* (GAC), que podem ser particularmente úteis para detecção de dano (ZANG; GRAFE; IMREGUN, 2001)

3.3 Modelos de Dano

Neste trabalho, entende-se por dano a degeneração estrutural que resulta em perda parcial de rigidez de um ou vários elementos da estrutura. Existem vários procedimentos propostos na literatura para simular dano (diminuição de rigidez) dentro de uma estrutura modelada com MEF. O mais simples deles, empregado nesta pesquisa, consiste na redução de rigidez de um ou vários elementos do MEF (PANDEY; BISWAS, 1991, 1994; SALAWU; WILLIAMS, 1994). O segundo modelo, também utilizado neste trabalho, para representar a diminuição de rigidez, é o apresentado por Sinha et. al. (2002), e está baseado numa aproximação simples da matriz de rigidez para um elemento de viga fissurado, supondo-se que a fissura permanece aberta durante todo o processo de carga. Para baixas frequências, esta abordagem modela a variação da rigidez local de forma acurada. Para as altas frequências, existem efeitos termodinâmicos, que o modelo anterior não leva em conta, fazendo com que perca precisão. A matriz de rigidez para um

elemento de secção retangular (bh), com uma fissura em seu interior (Figura 2), vem dada por (SINHA; FRISWELL; EDWARDS, 2002):

$$\left[K_{fiss} \right]_e = \begin{bmatrix} K_{11} & -K_{11} & K_{12} & K_{14} \\ -K_{11} & K_{11} & -K_{12} & -K_{14} \\ K_{12} & -K_{12} & K_{22} & K_{24} \\ K_{14} & -K_{14} & K_{24} & K_{44} \end{bmatrix} \quad (4)$$

onde:

$$K_{11} = \frac{H}{le^4} \left[2J + 3lf \left(\frac{2x_j}{le} - 1 \right)^2 \right] \quad K_{12} = \frac{H}{le^3} \left[J + lf \left(2 - \frac{7x_j}{le} + \frac{6x_j^2}{le^2} \right)^2 \right]$$

$$K_{14} = \frac{H}{le^3} \left[J + lf \left(1 - \frac{5x_j}{le} + \frac{6x_j^2}{le^2} \right)^2 \right] \quad K_{22} = \frac{H}{6le^2} \left[3J + 2lf \left(-2 + \frac{3x_j}{le} \right)^2 \right]$$

$$K_{24} = \frac{H}{6le^3} \left[3J + 2lf \left(2 - \frac{9x_j}{le} + \frac{9x_j^2}{le^2} \right)^2 \right] \quad K_{44} = \frac{H}{6le^3} \left[3J + 2lf \left(-1 + \frac{3x_j}{le} \right)^2 \right]$$

$$H = 12E(I_0 - I_{\bar{j}}), \quad x_j = \text{pos. fissura}$$

$$J = \frac{lf^3}{le^2}$$

I_0 = Inércia seção intacta

$$I_{\bar{j}} = \frac{b(h - \alpha_j)^3}{12}$$

$l_f = 1.5h$ comp. efetivo fissura

le = comprimento do elemento

α_j = altura da fissura

E = módulo de elasticidade

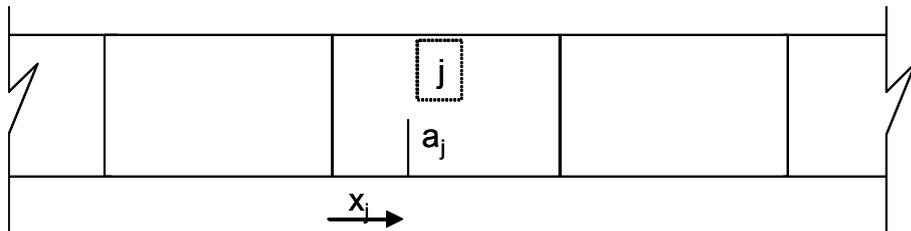


Figura 2. Posição (x_j) e tamanho (a_j) da fissura no elemento j da viga.

A expressão (4) foi obtida a partir do modelo de variação da rigidez que uma fissura induz num elemento de viga (Figura 3).

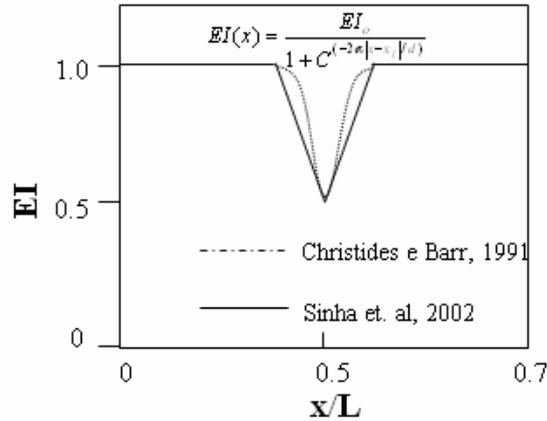


Figura 3. Comparação entre a variação de rigidez nas proximidades de uma fissura (SINHA; FRISWELL; EDWARDS, 2002)

A principal vantagem deste enfoque é que a matriz de rigidez da estrutura danificada pode ser escrita em função do tamanho e da posição do dano. Supondo-se que as rotações da viga não foram medidas experimentalmente, a ordem usual dos deslocamentos na Equação (4) foi alterada para $\{w1, w3, w2, w4\}$, com a finalidade de realizar a montagem da matriz de rigidez reduzida da estrutura, de acordo com o método de redução das equações de movimento proposto por Kidder (1973). O modelo considera que a fissura está contida dentro de um único elemento, porém, é possível obter as matrizes de rigidez para o caso da fissura estar localizada sobre mais de um elemento.

A matriz de rigidez da estrutura, $[K]_{fiss}$, pode agora ser montada segundo o procedimento normal empregado no MEF e vem dada pela expressão (5):

$$[K]_{fiss} = \sum_{e=1}^M [K]_e - [K_{fiss}]_e \quad (5)$$

onde $[K]_e$ é matriz de rigidez do elemento de viga intacto e $[K_{fiss}]_e$ é dado pela expressão (4).

A anterior abordagem dispensa o emprego de malhas de elementos finitos muito refinadas para modelar a variação de tensões na ponta da fissura, já que, a mudança na flexibilidade local pode ser incorporada de forma eficiente e rápida dentro dos elementos clássicos da estrutura, evitando também, o uso de superelementos (que contém termos singulares) e possibilitando sua aplicação em estruturas complexas. Outra vantagem desta abordagem é que o elemento pode ser usado em qualquer programa padrão de elementos finitos para analisar o comportamento dinâmico e estático da estrutura, possibilitando-se desta forma, seu uso em problemas de identificação de fissuras.

Outros modelos mais refinados apresentados por Bovsunovsky e Surace (2005) e Saavedra e Cuitiño (2001), utilizam a teoria da mecânica da fratura linear para modelar a variação de flexibilidade (rigidez) que uma fissura localizada introduz num elemento finito de viga. Outra opção é o desenvolvimento de malhas de elementos finitos em 2D ou 3D para vigas com fissuras (KRAWCZUK; OSTACHOWICZ, 1993), porém, esses modelos detalhados não apresentam uma melhora substancial para a detecção de dano devido à sua maior complexidade e alto custo computacional.

3.4 Equação de Movimento via Método dos Elementos Finitos (MEF)

O comportamento dinâmico de um sistema estrutural linear com n graus de liberdade pode ser representado pela seguinte equação matricial:

$$[M]\left\{u(t)\right\} + [C]\left\{\dot{u}(t)\right\} + [K]\left\{u(t)\right\} = f(t) \quad (6)$$

onde, $[M]$, $[C]$ e $[K] \in \mathfrak{R}^{n \times n}$ são as matrizes de massa, amortecimento e rigidez da estrutura; $\ddot{u}(t)$ é o vetor de acelerações; $\dot{u}(t)$ é o vetor de velocidades; $u(t)$ é o vetor de deslocamentos. O vetor $f(t) \in \mathfrak{R}^n$ é o vetor de forças de excitação. A Equação (6) é uma aproximação do comportamento dinâmico do sistema discretizado obtida, por exemplo, mediante o MEF. A partir de geometria e das propriedades dos elementos, as matrizes $[M]$ e $[K]$ podem ser calculadas. A presença da matriz de amortecimento se fundamenta em observações físicas e conveniência matemática. Mediante a incorporação de amortecimento viscoso é possível modelar a vibração transiente observada experimentalmente. Devido a que o amortecimento depende da estrutura total e não é uma propriedade intrínseca de seus membros, é quase impossível construir a matriz $[C]$ da mesma forma que são montadas as matrizes $[M]$ e $[K]$. Neste contexto, algumas hipóteses básicas para representar a matriz $[C]$, adotadas na literatura clássica, vêm dadas por:

$$[C] = \alpha[M] \quad (7)$$

$$[C] = \beta[K] \quad (8)$$

$$[C] = \alpha[M] + \beta[K] \quad (9)$$

onde, α e β são constantes de proporcionalidade que devem ser calculadas. As Equações (7) e (8) são pouco usadas na prática, já que, um melhor controle sobre o amortecimento modal pode ser alcançado utilizando a Equação (9), conhecida como amortecimento de Rayleigh. A Equação (9) simplesmente estabelece que o amortecimento está distribuído na estrutura, da mesma forma que a rigidez e a massa, o que é uma hipótese natural.

Como mencionado antes, é difícil quantificar os verdadeiros mecanismos do amortecimento em estruturas, por tanto, o modelo proporcional fornece uma maneira simples e matematicamente conveniente, (devido a sua propriedade de desacoplamento da Equação (6), através de uma transformação modal de coordenadas), para a análise de vibrações usando o MEF. Vale a pena salientar que, dependendo da natureza das matrizes coeficientes da Equação (6), o sistema pode ser classificado como linear (matrizes constantes) ou não linear (matrizes variáveis no tempo e / ou no espaço).

3.4.1 Problema do Auto Valor para Vibração Livre Não Amortecida

A equação para vibração livre não amortecida segundo a configuração dada nos termos do método dos elementos finitos pode ser obtida da Equação (6) fazendo-se $[C]=0$ e $f(t)=0$, ou seja:

$$[M]\{\ddot{u}(t)\} + [K]\{u(t)\} = 0 \quad (10)$$

A Equação matricial (10), por sua vez, tem solução vetorial da forma:

$$\{u(t)\} = \{q\} \text{sen}(\omega \cdot t + \delta) \quad (11)$$

onde, $\{q\}$ é um vetor arbitrário, ω é a frequência de vibração e δ é o ângulo de fase, sendo todos estes parâmetros incógnitas do problema. Substituindo a Equação (11) e sua segunda derivada na Equação (10) tem-se:

$$[[K] - \omega^2 [M]]\{q\} \text{sen}(\omega \cdot t + \alpha) = \{0\} \quad (12)$$

Como a Equação (12) deve ser válida para todos os valores de t , pode-se escrever:

$$[K]\{q\} = \omega^2 [M]\{q\} \quad (13)$$

A Equação (13) representa o problema do auto valor conhecido como problema de auto valor generalizado ou problema de auto valor linearizado. Em geral, para um sistema de n graus de liberdade, a solução da Equação (13) fornece N valores de ω^2 que são reais e positivos. Porém, se a matriz $[M]$ for singular, a Equação (13) terá um ou mais auto valores infinitos. Por outro lado, quando $[K]$ for singular, a Equação (13) terá um ou mais auto valores nulos. Associado a cada auto valor existe um auto vetor $\{q\}$. Na análise de vibrações as raízes quadradas dos autos-valores do problema de auto valor generalizado são chamadas de freqüências do sistema e os autos vetores são chamados de formas modais reais.

As propriedades de ortogonalidade para as formas modais normalizadas com relação à massa são:

$$\{\phi\}_i^T [M] \{\phi\}_j = \delta_{ij} \quad (14)$$

$$\{\phi\}_i^T [K] \{\phi\}_j = \omega_i^2 \delta_{ij} \quad (15)$$

onde, $\{\phi\}_i$ denota o i -ésima forma modal normalizada com relação à massa

com:

$$\begin{aligned} \delta_{ij} &= 0 & i \neq j \\ \delta_{ij} &= 1 & i = j \end{aligned}$$

Para solucionar o problema de auto valor dado pela Equação (13) existem diversos métodos numéricos que podem ser divididos em três grupos: métodos de transformação, métodos de iteração e métodos de busca de determinante.

Os Métodos de Transformação são empregados quando as matrizes envolvidas são de pequena ordem. Dentre esses métodos, merecem destaque o método generalizado de Jacobi, o método da transformação de Householder, o da transformação denominada QR e o denominado LR. É uma característica desses métodos, o fato de o cálculo envolver todos os autovalores e auto vetores do problema, sem possibilidade de tratamento separado.

Os métodos de iteração são métodos ideais para problemas envolvendo matrizes de ordem elevada, particularmente quando só são requeridos alguns poucos autovalores e auto vetores do problema. Entre estes, os métodos de iteração de Sub-espço e o de Lanczos, são os mais populares.

Os métodos de busca de Determinante caracterizam-se por calcular inicialmente as raízes da equação de frequência (polinômio característico). A avaliação dos coeficientes da equação de frequência demanda um grande número de operações numéricas, e, além disto, as raízes da equação de frequência se mostram muito sensíveis aos valores dos coeficientes. Assim, pequenos erros na avaliação destes coeficientes, levam a uma perda significativa de precisão nos valores das raízes.

Uma excelente apresentação destes três grupos de métodos para solução do problema de auto valor pode ser encontrada em Bathe (1996), Humar (1990) e Weaver e Johnston (1987).

3.4.2 Amortecimento Proporcional

Quando se trabalha com amortecimento proporcional, as formas modais são as mesmas do caso não amortecido e as frequências naturais

têm valores muito próximos dos valores do caso sem amortecimento. Por exemplo, para o modelo com amortecimento proporcional à rigidez, ou seja, $[C] = \beta_1 [K]$, pré-multiplicando-se e pós-multiplicando-se a matriz $[C]$ pela matriz $[\Phi]$ obtém-se: $[\Phi]^T \beta_1 [K] [\Phi] = \beta_1 [\Omega] = [c]$, onde $[c]$ é uma matriz diagonal que contém os coeficientes de amortecimento dos modos do sistema. O fato da matriz $[c]$ ser diagonal significa que as formas modais do sistema sem amortecimento são idênticas às formas modais do sistema amortecido. A mesma conclusão vale para o amortecimento de Rayleigh. O conceito de amortecimento de Rayleigh pode ser aplicado a qualquer número r de amortecimentos modais (ξ_j) e frequências naturais (ω_j) medidos. Neste caso, as constantes de proporcionalidade α_j (ver Equações (7), (8) e (9)), podem ser calculadas solucionando o seguinte sistema de equações simultâneas:

$$\frac{1}{2} \begin{vmatrix} 1/\omega_1 & \omega_1 & \omega_1^3 & \cdots & \omega_1^{2r-3} \\ 1/\omega_2 & \omega_2 & \omega_2^3 & \cdots & \omega_2^{2r-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1/\omega_r & \omega_r & \omega_r^3 & \cdots & \omega_r^{2r-3} \end{vmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_r \end{Bmatrix} = \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_r \end{Bmatrix} \quad (16)$$

Como mencionado anteriormente, as matrizes de amortecimento proporcional têm a propriedade de ser diagonalizadas por uma transformação de coordenadas modais, possibilitando o desacoplamento das n equações simultâneas de movimento dadas pela Equação (6).

3.4.3 Amortecimento Geral

Se a hipótese de amortecimento proporcional não é válida, por exemplo, no caso de amortecedores localizados, uma outra abordagem deve ser

seguida para o cálculo dos auto valores e formas modais da estrutura, pois elas dependerão da matriz de rigidez, da matriz de massa e da matriz de amortecimento do sistema estrutural. Para determinar as formas modais e freqüências neste caso pode-se proceder da seguinte forma.

Tomando-se como solução da Equação (6) o expresso na Equação (17), ou seja:

$$\{V\} = \{q\}e^{at} \quad (17)$$

e substituindo-se a Equação (17) e suas derivadas na Equação (6), chega-se em:

$$(a^2[M] + a[C] + [K])\{q\} = \{0\} \quad (18)$$

A Equação (18) terá solução não trivial para o vetor $\{q\}$, se e só se, o determinante da matriz coeficiente da Equação (19) for nulo, ou seja:

$$\det(a^2[M] + a[C] + [K]) = 0 \quad (19)$$

Assim sendo, a Equação (18) representa o Problema de Auto Valor Quadrático, e sua solução fornece $2N$ auto valores (a) e seus correspondentes auto vetores $\{q\}$. O problema pode ser transformado em um problema de auto valor *standard* ou em um problema de auto valor generalizado, permitindo-se assim o uso dos métodos de solução mencionados na seção 3.4.1, porém, a transformação implica em se dobrar a ordem das matrizes envolvidas; portanto, aumenta o esforço computacional da solução.

Para se obter a forma generalizada, primeiro define-se um vetor de tamanho $2N \times 1$, que contem as velocidades e deslocamentos desconhecidos, ou seja:

$$\eta = \begin{Bmatrix} \{\dot{V}\} \\ \{V\} \end{Bmatrix} \quad (20)$$

As equações de movimento escritas nos termos da forma ampliada (Equação (20)) ganham as seguintes redações:

$$[M]\{\dot{V}\} - [M]\{\dot{V}\} = 0 \quad (21)$$

$$[M]\{\ddot{V}\} + [C]\{\dot{V}\} + [K]\{V\} = 0 \quad (22)$$

Combinando-se as Equações (21) e (22) tem-se:

$$\begin{bmatrix} [0] & [M] \\ [M] & [C] \end{bmatrix} \begin{Bmatrix} \{\ddot{V}\} \\ \{\dot{V}\} \end{Bmatrix} + \begin{bmatrix} -[M] & [0] \\ [0] & [K] \end{bmatrix} \begin{Bmatrix} \{\dot{V}\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{0\} \end{Bmatrix} \quad (23)$$

ou ainda:

$$[A]\{\dot{\eta}\} = [B]\{\eta\} \quad (24)$$

onde:

$$[A] = \begin{bmatrix} [0] & -[M] \\ -[M] & -[C] \end{bmatrix} \quad (25)$$

$$[B] = \begin{bmatrix} -[M] & [0] \\ [0] & [K] \end{bmatrix} \quad (26)$$

A Equação (24) é chamada de equação de movimento reduzida, com as matrizes $[A]$ e $[B]$ sendo matrizes simétricas e de ordem $2N$, e tem solução da forma:

$$\{\eta\} = \{v\}e^{ft} \quad (27)$$

Substituindo-se a Equação (27) na Equação (24) tem-se:

$$[B]\{v\} = f[A]\{v\} \quad (28)$$

que é a equação do problema do auto valor generalizado, e cujos auto vetores e auto valores são dados pelo vetor $\{v\}$ e pelos valores f . Transformado-se a Equação (28) na forma *standard*, chega-se em:

$$\begin{bmatrix} -[M]^{-1}[C] & -[M]^{-1}[K] \\ [I] & [0] \end{bmatrix} \{v\} = f \{v\} \quad (29)$$

sendo que a Equação (29) representa o problema de auto valor *standard* com ordem igual a $2N$. Como a matriz apresenta coeficientes reais, os auto valores devem ser reais ou estar em pares complexo conjugados. Os auto vetores correspondentes aos auto valores complexos são também complexos, e aparecem em pares conjugados. Um auto vetor complexo é chamado de forma modal complexa. Vale assinalar que quase todos os sistemas estruturais são sub-amortecidos, e, neste caso, todos os auto valores são complexos e estão em pares conjugados com parte real negativa.

As propriedades de ortogonalidade, como nos casos anteriores, podem ser escritas como:

$$\{v\}_i^T [A] \{v\}_j = 0 \quad (30)$$

$$\{v\}_i^T [B] \{v\}_j = 0 \quad (31)$$

com $f_i \neq f_n$.

3.4.4 Funções de Resposta em Freqüência Analíticas (FRFs)

Supondo-se que o sistema dado pela Equação (6) é excitado por um conjunto de forças senoidais com freqüência ω , porém, com diferentes amplitudes e fases, então a equação de movimento fica:

$$[M] \left\{ \ddot{u}(t) \right\} + [C] \left\{ \dot{u}(t) \right\} + [K] \left\{ u(t) \right\} = f_0 e^{-i\omega t} \quad (32)$$

onde, f_0 é um vetor $n \times 1$ que contem as amplitudes da excitação; t é o tempo e e é a base dos logaritmos naturais. Assumindo que a solução da Equação (32) vem dada por:

$$\{u(t)\} = z e^{-i\omega t} \quad (33)$$

na qual z é um vetor de deslocamentos de tamanho $nx1$ contendo amplitudes complexas independentes do tempo. Substituindo a Equação (33) na Equação (32) se obtém a seguinte solução:

$$z = \left[[K] - \omega^2 [M] - i\omega [C] \right]^{-1} f_o \quad (34)$$

A matriz de receptância é definida como:

$$[R] = \left[[K] - \omega^2 [M] - i\omega [C] \right]^{-1} \quad (35)$$

O elemento ij da matriz de receptância representa a resposta do i -ésimo grau de liberdade, quando uma excitação é aplicada no j -ésimo grau de liberdade. Cada elemento da matriz $[R]$ é função de ω e é chamado de função de resposta em frequência (FRF). Conseqüentemente, a matriz $[R]$ é uma matriz de FRF. Para fins de detecção de dano, o custo computacional de obter a matriz $[R]$ a partir da Equação (35) é muito alto, devido a que, para cada frequência, no domínio de interesse, uma matriz nxn deve ser invertida. Utilizando a técnica da decomposição modal e supondo que o amortecimento é proporcional, pode mostrar-se que o ij elemento da matriz $[R]$ vem dado por:

$$R_{ij} = \sum_{m=1}^n \left[\frac{1}{-\omega^2 - 2i\omega\xi_m\omega_m + \omega_m^2} \right] \phi_{im}\phi_{jm} \quad (36)$$

onde, ω_m é a m -ésima frequência natural, ξ_m é o amortecimento modal do modo m , ϕ_{im} e ϕ_{jm} são elementos da matriz de formas modais (da estrutura sem amortecimento) e n é o número de graus de liberdade. O esforço computacional para calcular a matriz $[R]$ usando a Equação (36) é consideravelmente menor do que o empregado com a Equação (35). As

matrizes de aceleração e mobilidade podem ser facilmente calculadas a partir da matriz de receptância. Como esperado, a matriz $[R]$ é simétrica.

3.4.5 Introdução de Dano no MEF da Estrutura

O método apresentado nesta pesquisa assume que o dano afeta a rigidez da estrutura, porém, o amortecimento e/ou a massa poderiam também ser considerados dentro da estratégia proposta. Para introduzir dano na estrutura é utilizado um parâmetro escalar de dano que afeta as propriedades de cada elemento finito do modelo ou, de forma alternativa, um elemento finito fissurado pode ser introduzido no MEF (ver Equação (4)). O objetivo da avaliação da integridade estrutural é calcular um vetor, que contém os parâmetros de dano, para todos os elementos da estrutura. A rigidez do elemento danificado pode ser calculada segundo a seguinte expressão:

$$[K]_{dan}^i = \alpha_i [K]^i \quad i = 1, \dots, ne \quad \text{com } ne = \text{número de elementos finitos} \quad (37)$$

onde, $[K]_{dan}^i$ é a matriz de rigidez do *i*-ésimo elemento finito danificado da estrutura, $[K]^i$ é a matriz de rigidez do *i*-ésimo elemento finito intacto, α_i é o *i*-ésimo parâmetro de dano correspondente ao *i*-ésimo elemento finito. Os valores do parâmetro α_i estão contidos no intervalo $[0, 1]$, porém aumentos de rigidez, produzidos, por exemplo, por reforço, podem ser facilmente incorporados. Estes valores garantem que a posição do dano identificada é única e que seu valor (redução e/ou aumento) tenha sentido físico. Utilizando qualquer dos dois modelos de danificação dentro das Equações (6) ou (35), dados analíticos de vibração podem ser obtidos.

A implementação das simulações numéricas, para obtenção das respostas dinâmicas usadas nos exemplos apresentados nos capítulos 7 e 8, foi realizada na linguagem FORTRAN.

3.5 Uso de Dados Experimentais para Avaliação da Integridade Estrutural

A informação experimental pode ser usada para avaliar a integridade das estruturas, assim como, para aprimorar o conhecimento e o entendimento sobre o seu comportamento. Isso é alcançado mediante a observação da resposta do sistema ante um conjunto de excitações e condições de contorno conhecidas. Na atualidade, a análise modal é a técnica mais usada para realizar testes dinâmicos (EWINS, 1984). Esta técnica é usada para obter um modelo experimental da estrutura, no qual, é descrito o comportamento dinâmico através de um conjunto de frequências naturais, formas modais e fatores de amortecimento. Esta informação é obtida mediante testes, que podem ser de vibração forçada, de vibração livre e de vibração ambiental. Os dois primeiros métodos estão limitados a estruturas relativamente pequenas. No caso de estruturas como pontes ou plataformas marítimas, os altos custos e riscos destes ensaios os tornam inviáveis. Uma alternativa atraente é o uso de vibração ambiental, induzida pelo tráfego, o vento ou por um abalo sísmico.

A técnica de análise modal usando vibração ambiental tem sido aplicada com sucesso em diversas estruturas (REN; ZHAO; HARIK, 2004; ZONG et. al., 2005). Dentre a ampla variedade de métodos reportados para obter parâmetros modais a partir de vibração ambiental (ver, por exemplo, os

trabalhos de ANDERSEN; BRINCKER; KIRKEGAARD, 1996; JAMES; CARNE; LAUFFER, 1995), o método da identificação estocástica do subespaço (VAN OVERSCHEE; De MOOR, 1996) é, talvez um dos mais avançados neste grupo.

Neste trabalho, se supõe que os dados de vibração são tomados diretamente da estrutura, sem hipótese nenhuma sobre ela, e por tanto, são considerados como referência para o MEF. Porém, limitações e erros na informação experimental podem aparecer. Geralmente, não é possível medir alguns graus de liberdade (por exemplo, rotações), as medidas realizadas estão contaminadas com certo nível de ruído e o número de modos identificado está limitado às baixas frequências.

3.5.1 Funções Objetivo

Nesta pesquisa, o problema de detecção de dano é definido como um problema de programação matemática. Nessa abordagem é necessário definir uma função objetivo (f_{obj}) para determinar a diferença (resíduo) entre a predição numérica e o comportamento real da estrutura. Para fins de detecção de dano, o resíduo deve ser sensível a pequenas variações locais da estrutura. Basicamente, uma combinação de frequências naturais e formas modais constituem uma função objetivo sensível para detecção de dano.

Para os métodos de detecção de dano é importante expressar o valor da rigidez do sistema em função das variáveis α_i ($0 < \alpha_i < 1$) dadas pela Equação (37) ou dos parâmetros que caracterizam o dano dentro de um elemento finito da estrutura (ver Equação (4)), que constituem as variáveis

do problema de otimização, e do valor inicial da rigidez dos N componentes (elementos finitos) que conformam a estrutura.

A matriz de rigidez global da estrutura pode ser expressa como:

$$[K] = \sum_{i=1}^N \alpha_i [K]^i \quad (38)$$

onde, $[K]^i$ é a matriz de rigidez expandida do i -ésimo elemento do sistema. Supondo que o modelo de elementos finitos é correto (o modelo analítico inicial é uma boa representação da estrutura original) e que as frequências $\omega_{\text{exp } j}$ e formas modais $\{\phi_j\}_{\text{exp}}$, determinadas experimentalmente, estão disponíveis tem-se, a partir das Equações (13) e (38), que o valor do vetor de força residual é (MARES; SURACE, 1996):

$$\{F_r\}_j = \sum_i^N \alpha_i [K]_i \{\phi_j\}_{\text{exp}} - \omega_{\text{exp } j}^2 [M] \{\phi_j\}_{\text{exp}} \quad (39)$$

O vetor de força residual $\{F_r\}_j$ será nulo caso os dados experimentais estejam livres de erros sistemáticos e de erros aleatórios de medição e, portanto, os valores α_i serão todos iguais a um (a matriz de rigidez da estrutura é a matriz verdadeira). Caso contrário, o vetor $\{F_r\}$ será diferente de zero e os parâmetros que minimizam a função objetivo dada pela Equação (40) devem ser calculados para realizar a identificação do dano.

$$f_{\text{obj}} = \sum_i^p \{F_r\}_i^T \bullet \{F_r\}_i \quad (40)$$

O somatório entre i e p , na Equação (40), representa o número de modos usados no cálculo da função objetivo. A Equação (38) garante que a simetria, a esparcidade e a informação sobre conectividade contida na matriz de rigidez sejam mantidas dentro do problema de otimização formulado. Por

outro lado, as variáveis α_i devem estar contidas no intervalo (0,1) para evitar que ajustes pouco realísticos, como elementos com rigidez negativa, apareçam durante a análise. Adicionalmente, a Equação (40) emprega só os modos experimentais e, portanto, não requer do cálculo de correlação com dados analíticos (por exemplo, do COMAC).

Uma outra expressão para a função objetivo, extraída da área de identificação de sistemas e utilizada para detecção de dano está definida na Equação (41) (MOSLEM; NAFASPOUR, 2002):

$$f_{obj} = \sum_{i=1}^L \sum_{j=1}^C (\varphi_{ij}^e - \varphi_{ij}^a)^2 + \sum_{i=1}^L \left[1 - \frac{\omega_i^e}{\omega_i^a} \right] \quad (41)$$

onde, φ_{ij}^e é o j -ésimo elemento do i -ésimo auto vetor determinado experimentalmente, φ_{ij}^a é o j -ésimo elemento do i -ésimo auto vetor determinado analiticamente e ω_i^e e ω_i^a são as i -ésimas freqüências natural experimental e analítica, respectivamente. L é o número de auto vetores / auto valores medidos e C é o número de elementos dos auto vetores medidos. A minimização da função objetivo fornece, por exemplo, os valores dos parâmetros α_p e x_p que definem a altura e posição da fissura dentro da estrutura (ver Equação (4)), com $p=1, \dots, N$ e N é o número de elementos finitos utilizados. A Equação (41) pode ser escrita da seguinte forma:

$$f_{obj}(A) = F_{ob}(\alpha_p, x_p) \quad (42)$$

onde, $A = \{ \alpha_p, x_p \}$ e $\alpha_p, x_p \in \mathfrak{R}^+$

A vantagem principal desta abordagem é que o conjunto completo de dados modais não precisa ser medido, já que, o cálculo da função objetivo envolve só a diferença entre componentes destes vetores.

Uma regra geral para o uso de FRFs em detecção de dano é que os pontos usados estejam na vizinhança das freqüência naturais, mas não exatamente sobre elas, já que, as FRFs não são muito acuradas nestes pontos. Uma função objetivo que pode ser empregada para identificação de dano e que expressa a divergência entre a FRF medida e a FRF analítica, supondo que existe uma única excitação, pode ser calculada mediante a seguinte equação:

$$f_{obj} = \sum_{k=k_1}^{k_L} \int_{\omega_1}^{\omega_2} \left| [R(\Omega)]_{jk}^m - [R(\Omega)]_{jk}^a \right| d\omega \quad (43)$$

Onde, k é o grau de liberdade onde a resposta é medida, j indica a posição da excitação, $[R(\Omega)]_{jk}^m$ é a jk -ésima FRF de receptância medida, $[R(\Omega)]_{jk}^a$ é a jk -ésima FRF analítica (ver Equação (36)), K_L é o número total de graus de liberdade onde as medidas foram realizadas, $/ /$ indica magnitude complexa e ω_1 e ω_2 definem o intervalo de integração. No caso de múltiplas excitações e múltiplas respostas medidas, a função objetivo pode ser definida como:

$$f_{obj} = \sum_{j=j_1}^{j_q} \left[\sum_{k=k_1}^{k_L} \int_{\omega_1}^{\omega_2} \left| [R(\Omega)]_{jk}^m - [R(\Omega)]_{jk}^a \right| d\omega \right] \quad (44)$$

onde, j_q indica o grau de liberdade onde a excitação é aplicada. As FRFs são função da rigidez do sistema e, portanto, são também função dos parâmetros de dano α_i .

Por outro lado, o uso de respostas no domínio do tempo permite o estudo direto das características não lineares como parte do processo de identificação de dano, já que, estas respostas capturam com maior fidelidade o comportamento físico do sistema, sob a condição de que os registros temporais sejam suficientemente longos para caracterizar as diferencias

entre as respostas da estrutura sadia e a danificada. Uma função objetivo que pode ser usada neste contexto e a proposta por Banks et. al. (1996):

$$f_{obj} = \sum_{i=1}^m p_i \sum_{j=1}^t |x_{ie}(t_j) - x_{in}(t_j)|^2 \quad (45)$$

Onde, $x_{ie}(t_k)$ e $x_{in}(t_k)$ são as m respostas experimentais e numéricas, respectivamente, medidas em um total t de t_j tempos discretos e p_i são fatores de peso introduzidos para levar em conta diferenças de amplitude das respostas perante diferentes excitações. A função objetivo dada pela Equação (45) tem múltiplos pontos ótimos, o qual faz com que a solução dependa do ponto inicial de busca, quando empregados métodos baseados em gradientes. Para simular a presença de ruído nas medições usadas na definição da função objetivo, pode ser adicionado ruído aleatório normalmente distribuído.

Finalmente, vale salientar que o sucesso de qualquer procedimento de identificação de dano depende fortemente da quantidade de informação disponível sobre a área danificada da estrutura. Devido a que, esta área não é conhecida *a priori*, é vantajoso usar a maior quantidade possível de informação disponível sobre estrutura inteira.

3.6 Análise de Resultados

A hipótese de comportamento linear da estrutura é empregada nesta pesquisa. Porém, isto não implica de forma alguma que a estrutura não possa sofrer deformações irreversíveis (danos). Por exemplo, analisando o histórico de carga de uma conexão estrutural depois de vários ciclos de carga, pode-se perceber que ela retorna a uma nova posição de equilíbrio, a

qual pertence também à zona elástica e, portanto, o modelo elástico continua sendo válido. Por outro lado, o fenômeno de escoamento por si só pode não influenciar a rigidez e as características de vibração de estrutura. Voltando ao exemplo da conexão, a rigidez do material não muda na nova posição de equilíbrio e conseqüentemente, o procedimento de detecção não conseguira captar sua presença (escoamento). Para que o dano seja detectável pelos métodos aqui estudados é preciso que exista uma mudança de secção transversal ou de momento de inércia (por exemplo, causada pela presença de fissuras). Para algumas estruturas de concreto, submetidas a carga cíclica, é possível que a nova posição de equilíbrio não seja paralela ao módulo de elasticidade original e, neste caso, o dano pode ser detectado. Em ligações rebitadas, também é possível detectar a perda de rigidez da união, ocasionada pela fluência de um rebite.

Infelizmente, não existe uma única resposta que possa ser medida e que seja capaz de identificar todos os tipos de dano que venham a acontecer dentro de uma estrutura. Na atualidade, não existe uma técnica universal para DD reconhecida pela comunidade científica especializada. Confiar cegamente numa técnica exclusiva para identificar dano é inviável; pelo contrário, uma combinação das varias técnicas existentes pode fornecer maior confiabilidade na avaliação da integridade estrutura.

4 Algoritmos Heurísticos para Detecção de Dano

4.1 Introdução

Neste trabalho, o problema inverso DD baseado em modelos é definido como um problema de programação não linear, no qual se estabelece uma Função Objetivo (f_{obj}) para comparar dados experimentais do sistema com dados simulados do modelo (ver Equações 40 a 45), e, mediante sua minimização / maximização, são calculados os parâmetros que identificam o dano.

Dentre as técnicas heurísticas desenvolvidas recentemente, e abordadas neste capítulo, o algoritmo *Particle Swarm Optimization* (PSO) proposto por Kennedy e Eberhart (1995), o algoritmo *Simulated Annealing* em suas varias versões (por exemplo, a versão de Kirkpatrick, Gelatti e Vecchi 1983, a de Corana et. al., 1987 ou a de Koh e Liaw ,2003) se apresentam, junto com os AGs, como opções promissoras para atacar o problema de detecção de dano. Esta asseveração é fundamentada em diversos fatores, entre os quais pode-se mencionar:

- sua facilidade de programação e de formulação;
- sua capacidade de fazer uso eficiente de um grande número de processadores;

- não requerem continuidade na definição do problema de otimização e não depende da estimativa de um ponto inicial para garantir a convergência (para o problema DD, uma estimativa *a priori* para começar a busca é quase impossível, devido a grande quantidade de opções);
- estão adaptados para procurar soluções globais o quase globais.

Adicionalmente, como mostrado por Schutte e Groenwold (2005), o algoritmo PSO ultrapassa o desempenho dos AGs em vários problemas difíceis de programação não linear, notadamente, em problemas de otimização sem restrições.

Por outro lado, é um fato conhecido que os algoritmos heurísticos precisam de um maior número de avaliações da função objetivo, para encontrar uma solução ótima, quando comparados com as técnicas baseadas em gradientes e/ou derivadas de segunda ordem. Porém, a principal desvantagem das técnicas clássicas de programação não linear, quando aplicadas ao problema DD, é que elas são suscetíveis de convergir para ótimos locais e, portanto, não podem ser empregadas com sucesso em problemas não lineares com múltiplos pontos ótimos, como os estudados neste trabalho. Além desta limitação, os algoritmos clássicos são altamente sensíveis à presença de ruído nos dados experimentais (HEMEZ et. al., 1995). Isto se deve ao fato de que os gradientes usados dependem explicitamente dos vetores modais medidos e/ou da sua expansão (quando utilizados dados modais), desta forma, qualquer erro de medição ou de modelagem irá corromper a qualidade dos gradientes. De maneira geral, a mesma conclusão pode ser feita para qualquer método baseado em sensibilidades (gradientes).

Recentemente, diversas estratégias para evitar a convergência prematura e melhorar a capacidade de busca dos métodos heurísticos básicos vêm sendo propostas. Por exemplo, Koh e Liaw (2003), apresentam uma das primeiras tentativas para melhorar o desempenho numérico de um sistema de identificação estrutural baseado em Algoritmos Genéticos (AGs) mediante a combinação do AG com um método de busca local. Outra possibilidade estudada por Hwang e He (2006), para solucionar problemas de otimização estrutural e problemas de identificação de sistemas, é a combinação de operadores do *Simulated Annealing* (SA) com os AGs com o intuito de melhorar a convergência do AG. O método se baseia em um algoritmo genético de parâmetros reais com um operador de *crossover* novo e incorpora as vantagens que o SA possui (capacidade de fugir de pontos locais) melhorando desta forma o desempenho global do método.

Finalmente, o problema DD apresenta três características que tornam interessante o uso de algoritmos heurísticos para sua solução. A primeira, é que se trata de um problema que deve ser solucionado em presença de ruído. A segunda, é que a detecção de dano baseada em modelos é um problema fortemente não linear, onde podem existir múltiplos pontos ótimos e pode ser de natureza descontínua (SINHA; FRISWELL; EDWARDS, 2002; BLAND; KAPANIA, 2004). Por último, a estratégia baseada em heurísticas não precisa um conjunto completo de medidas, por exemplo, medidas em todos os graus de liberdade. As anteriores razões dificultam muito o uso de técnicas clássicas de otimização para resolver o problema DD. A tabela 1 mostra uma comparação entre métodos tradicionais e algoritmos heurísticos para solução de problemas de detecção de dano.

Tabela 1. Comparação entre algoritmos clássicos e algoritmos heurísticos.

| Heurísticos | Clássicos |
|--|--|
| Soluções globais | Soluções locais |
| Não requerem cálculo de derivadas | Cálculo de derivadas necessário |
| Não dependem de ponto inicial | Dependem de ponto inicial |
| Sem restrições: continuidade, convexidade | Restrições: Continuidade, convexidade |
| Pouca sensibilidade a ruídos | Sensíveis a ruídos |
| Maior número de avaliações da função objetivo | Menor número de avaliações da função objetivo |
| Operações lógicas básicas | Programação dependente do problema |
| Uso de regras heurísticas | Embasamento matemático |
| Sem versões comerciais; literatura fragmentada. | Com versões comerciais; livros publicados. |

4.2 Algoritmo *Simulated Annealing* (SA)

O algoritmo SA se baseia na analogia entre o processo de resfriamento lento de sólidos e a solução de problemas de otimização de grande porte, com variáveis contínuas e discretas. Foi proposto por Kirkpatrick, Gelatti e Vecchi em 1983. O SA tem sido empregado no campo da física e da cristalografia, para ajustar modelos atômicos de proteínas usando dados experimentais e informações químicas (BRUNGER, 1991). Uma das primeiras aplicações do SA, para o posicionamento ótimo de sensores e

atuadores em estruturas espaciais, foi feita por Salama et al. (1990), e sua aplicação é relativamente recente na disciplina de otimização estrutural, onde os objetivos principais são obter estruturas com formas, pesos, resistências ótimas e/ou controlar parâmetros de vibração de diversos sistemas (GENOVESE; LAMBERTI; PAPPALETTERE, 2005; KINCAID, 1992).

No trabalho apresentado por Bennage e Dhingra (1995), é mostrada a robustez do SA mediante sua aplicação em problemas de otimização multi-objetivo para o projeto de treliças com variáveis contínuas e discretas. Embora nas áreas anteriores o algoritmo SA tenha sido usado com freqüência, poucos trabalhos têm sido publicados aplicando esta metodologia na área de dinâmica de estruturas, nos campos de ajuste de modelos e detecção de dano (ZIAEI-RAD, 2005; BEGAMBRE; LAIER, 2006; ZHOU; KIM; YANG, 2005; BEGAMBRE; LAIER, 2005). Uma contribuição deste trabalho é testar o algoritmo *Simulated Annealing* em problemas de detecção de dano. Para este fim, a variante do SA apresentada por Corana et. al. (1987), é avaliada em diversas funções teste e em casos de detecção de dano simulados.

O processo de *annealing* (recozimento) consiste em aquecer uma substancia até que ela alcança um determinado nível de energia e, seguidamente, reduzir de forma lenta sua temperatura. Mediante este procedimento, se permite à substancia atingir o equilíbrio térmico em cada temperatura. Eventualmente, a temperatura decresce até que o material “congela”. Se a temperatura é diminuída de forma suficientemente lenta o processo de *annealing* sempre atinge o estado de mínima energia a partir de um número quase infinito de estados iniciais possíveis. O *annealing* é um

processo natural de otimização e sua simulação na área de matemática aplicada é conhecida como *Simulated Annealing*.

O SA é, basicamente, um procedimento de busca aleatória de pontos ótimos globais, que permite movimentos para fugir de pontos ótimos locais. Estes movimentos de fuga são controlados através do conhecido critério de Metropolis (METROPOLIS et. al., 1953). As vantagens que tornam atraente o SA para a solução de problemas inversos são sua não dependência do cálculo de gradiente da função objetivo e sua capacidade de fugir de pontos ótimos locais. O procedimento geral, para otimizar uma função empregando o SA, envolve a definição de uma função objetivo, a proposta de um mecanismo para gerar variações na configuração atual, o estabelecimento de um programa de esfriamento e, finalmente, a fixação de um critério de parada. Estes passos fundamentais são descritos em detalhe nas seguintes seções.

4.2.1 Função Objetivo

A expressão para a função objetivo utilizada para detecção de dano mediante o SA pode ser uma das dadas pelas Equações (40) a (45). O problema agora, pode ser formulado da seguinte maneira: achar A_{ot} que satisfaça a Equação (46):

$$f_{obj}(A_{ot}) = \min\{F(A) / A \in \mathfrak{R}^n\} \quad (46)$$

onde, o estado do sistema é definido pelo vetor de configuração atual A (parâmetros de dano).

4.2.2 Estruturas de Vizinhança

Para definir um mecanismo gerador de variações aleatórias para a configuração atual é necessário estabelecer uma estrutura de vizinhança. Este mecanismo é uma forma de perturbar A para obter novas configurações \bar{A} . Entre os tipos de vizinhança sugeridos na literatura se podem mencionar o ajuste unidirecional aleatório (CORANA et. al., 1987), o ajuste de raio fixo, o ajuste normal e o ajuste de Cauchy. Os três últimos precisam da definição de um parâmetro extra para determinar o ponto vizinho (o raio, no caso do ajuste de raio fixo e as variâncias, para o ajuste normal e de Cauchy). Por este motivo, o mecanismo de transição de ajuste unidirecional aleatório foi empregado neste trabalho. Esta estrutura de vizinhança funciona como descrito a seguir.

A avaliação da função f_{obj} é feita no ponto inicial A_k e seu valor $f_{obj}(A_k)$ é guardado. Um novo ponto \bar{A}_j é determinado mediante uma variação aleatória introduzida no elemento α_i do vetor A_k :

$$\bar{\alpha}_i = \alpha_i + \lambda \cdot \eta_i \quad (47)$$

na Equação (47), λ é um número aleatório no intervalo (-1,1), η_i representa um elemento do vetor η , que é o comprimento do passo para o vetor A_k e α_i , $\bar{\alpha}_i$ são elementos de A_k e \bar{A}_j , respectivamente. Seguidamente, o valor da função $f_{obj}(\bar{A}_j)$ é calculado. Se $f_{obj}(\bar{A}_j) < f_{obj}(A_k)$, \bar{A}_j é aceite e A_k é substituído por \bar{A}_j , portanto, o algoritmo “desce” (se aproxima do mínimo).

Se $f_{obj}(\bar{A}_j) > f_{obj}(A_k)$, a probabilidade de \bar{A}_j ser aceito vem dada pelo critério de Metropolis:

$$P_r[\bar{A}_j] = e^{\left(\frac{\Delta f_{obj}}{T}\right)} \quad (48)$$

na Equação (48), $\Delta f_{obj} = f_{obj}(A_k) - f_{obj}(\bar{A}_j)$ e T é o parâmetro de “temperatura”, que é o análogo da temperatura no processo físico de *annealing* (recozimento). Na prática, o valor P_r é comparado com P , que é um número randômico no intervalo (0,1). Se $P_r > P$, o novo ponto é aceito e A_k é substituído por \bar{A}_j e o algoritmo “Sobe” (se afasta do mínimo), caso contrário, \bar{A}_j é rejeitado. O algoritmo SA começa numa temperatura “alta” T_0 e uma seqüência de pontos \bar{A}_j é gerada até atingir o equilíbrio, isto é, se obtém uma seqüência de pontos \bar{A}_j , cujo valor médio de f_{obj} atinge um valor estável à medida que j aumenta. O melhor ponto alcançado (mínimo nesta temperatura) é guardado como A_{ot} .

4.2.3 Programa de Esfriamento

Neste estágio do processo, o parâmetro de controle T é diminuído de acordo com uma regra de decremento, conhecida como programa de esfriamento, que obedece à relação:

$$T_{j+1} = \theta \cdot T_j \quad (49)$$

onde, θ é uma constante real cujo valor está no intervalo (0,1). Com o valor de T reduzido segundo a Equação (49), uma nova seqüência de pontos é gerada a partir de A_{ot} , até que o novo equilíbrio seja atingido e o processo

continua até o critério de parada ser satisfeito. É importante ressaltar que, para finalizar o programa de esfriamento, o algoritmo precisa ter realizado um número predeterminado de iterações na mesma temperatura. Corana et. al. (1987) recomendam escolher o valor máximo entre 100 e $5N$, onde N é o número de variáveis do problema estudado.

O valor de temperatura inicial depende da função que vai ser otimizada e da definição da vizinhança empregada no algoritmo. Um critério usado para definir este parâmetro é a taxa de aceitação, definida como o número inicial de avaliações da função objetivo aceitas (movimento de descida / subida) sobre o número total de avaliações realizadas (número total de movimentos). Na prática, um valor de temperatura inicial deve ser tal que o valor da taxa de aceitação fique entre 0.5 e 0.9. Se o valor da taxa for maior que 0.9, uma percentagem significativa de avaliações é gasta num estado “derretido”, desperdiçando esforço computacional numa procura equivalente a uma busca aleatória. Se a taxa de aceitação for menor que 0.5, a probabilidade do algoritmo ficar preso num ótimo local aumenta.

4.2.4 Critério de Parada

O algoritmo termina para um valor de T pequeno, para o qual, nenhuma melhora no valor de f_{obj} , possa ser esperada. Na prática, o critério de parada pode ser definido mediante um valor de tolerância da seguinte forma: se a diferença entre valores finais da função objetivo das p últimas temperaturas e o valor atual da função for menor que o valor da tolerância, o algoritmo termina. A listagem do programa SA em FORTRAN empregado

neste estudo é apresentada no Apêndice A. A Figura 4 apresenta o fluxograma básico do SA.

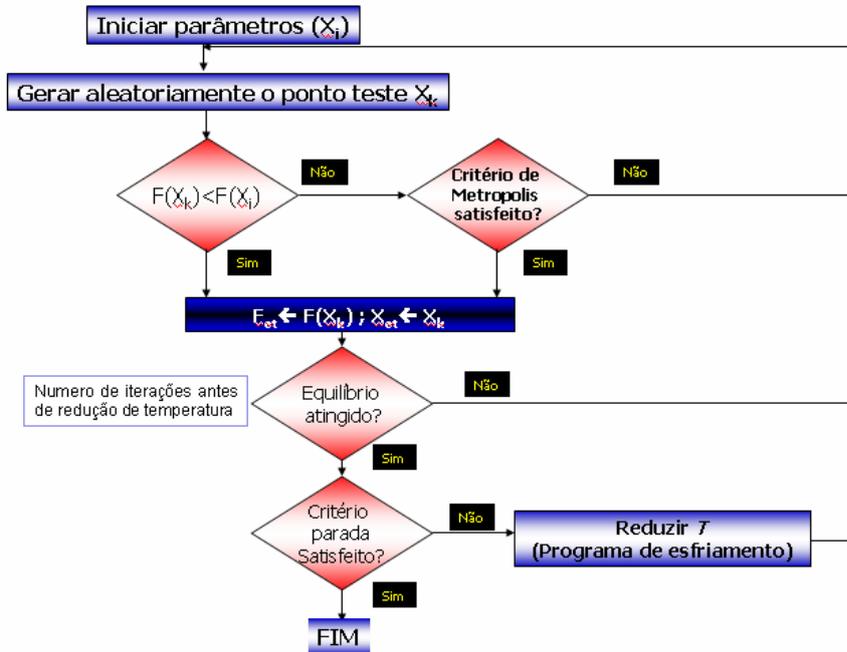


Figura 4. Diagrama de fluxo SA básico.

Da anterior exposição, fica claro que, grandes diferenças no valor da função e/ou baixas temperaturas diminuem a probabilidade de um movimento de “subida” do algoritmo. Por outro lado, a variação do passo η é feita segundo o proposto em Corana et. al. (1987), de tal forma que, a relação entre o número de movimentos de subida/descida aceitos e o total de movimentos realizados, permaneça próxima de 0.5, e, de esta maneira, manter um bom seguimento do valor da função estudada. Finalmente, a estratégia de realizar uma busca unidirecional aleatória tem melhores chances de manter a direção favorável de busca nas iterações seguintes e de se aproximar do ponto ótimo de forma sistemática. A perturbação simultânea, além de não manter a direção favorável de busca, também não

fornece informação sobre o efeito de cada variável na violação de restrições em problema com restrições.

4.3 Algoritmo *Particle Swarm Optimization* (PSO)

O algoritmo PSO foi proposto por Kennedy e Eberhart em 1995 e está baseado na simulação de um modelo de interação social simplificado. Sabe-se que compartilhar informação no ambiente social oferece aos membros do grupo uma vantagem evolutiva. Esta observação é fundamental para o desenvolvimento do PSO.

O PSO é membro da ampla categoria de métodos conhecidos como *Swarm Intelligence*, utilizados para resolver problemas de programação não linear. Desde sua introdução, muitas aplicações em otimização estrutural e multidisciplinar tem sido publicadas. Uma revisão das aplicações do PSO, para solucionar problemas de otimização na área de sistemas elétricos, é apresentada em Alrashidi e El-Hawary (2006). Outras aplicações para o projeto térmico ótimo de prédios podem ser encontradas em Wetter e Wright (2004). Aplicações adicionais para otimização estrutural (forma e peso) são apresentadas em Venter e Sobieszczanski-Sobieski (2003), Schutte e Groenwold (2003) e Fourie e Groenwold (2002).

Por outro lado, nenhum estudo sobre aplicações e modificações do PSO para detecção de dano tem sido reportado. Desta forma, esta pesquisa representa a primeira tentativa de avaliar o desempenho numérico do algoritmo, assim como, de melhorar seu comportamento, em problemas de identificação de dano, mediante a estratégia híbrida desenvolvida.

No PSO existem vários parâmetros explícitos cujos valores afetam a maneira na qual o algoritmo faz a busca no espaço do problema (KENNEDY, 1997; ALRASHIDI; EL-HAWARY, 2006; VENTER; SOBIESZCZANSKI-SOBIESKI, 2003; SCHUTTE; GROENWOLD, 2003; FOURIE; GROENWOLD, 2002). Estes parâmetros heurísticos controlam as propriedades de convergência do PSO, portanto, mediante a correta seleção destas constantes, o desempenho do algoritmo pode ser melhorado, como explicado nas seguintes seções.

4.3.1 Mecanismo e Parâmetros Básicos do PSO

Neste trabalho, é implementada a versão global assíncrona do PSO com redução de inércia linear (SCHUTTE; GROENWOLD, 2003). O algoritmo básico parte da hipótese de que cada partícula (solução candidata) da população (*swarm* ou conjunto de N partículas) voa sobre o espaço de busca, procurando por regiões promissoras da paisagem (por exemplo, em um problema de maximização), regiões que possuam valores da função objetivo maiores que outros, descobertos previamente. Neste contexto, a posição de cada partícula é ajustada, utilizando a informação social compartilhada pelos membros do enxame (*swarm*), e cada partícula tenta mudar sua posição a um ponto onde, ela e o enxame, tinham um valor maior da função objetivo em iterações prévias.

As partículas são manipuladas de acordo com as seguintes equações vetoriais (SHI; EBERHART, 1998):

$$p_{k+1}^i = p_k^i + v_{k+1}^i \tag{50}$$

$$v_{k+1}^i = \omega v_k^i + C_1 rand_1 (b_k^i - p_k^i) + C_2 rand_2 (b_k^g - p_k^i) \quad (51)$$

onde, k indica um incremento pseudo-temporal unitário, p_k^i representa a posição de cada partícula i (soluções candidatas) no tempo (iteração) k , p_{k+1}^i é a posição da partícula i no tempo $k+1$, b_k^i representa a melhor posição alcançada pela partícula i no tempo k (melhor posição individual), b_k^g é a melhor posição no enxame no tempo k (melhor posição atingida pela partícula usada para guiar as outras partículas no enxame), v_k^i é a velocidade da partícula i no tempo k e v_{k+1}^i é a velocidade ajustada da partícula i no tempo $k+1$. Todos os vetores nas Equações (50) e (51), são de dimensão $m \times 1$, onde m é o número de parâmetros otimizados, $rand1$ e $rand2$ são números aleatórios independentes (com probabilidade uniforme) entre 0 e 1. Os parâmetros C_1 e C_2 controlam o fluxo de informação entre o enxame atual. Se $C_2 > C_1$, então a partícula põe mais confiança no enxame, de outra forma, a partícula põe mais confiança nela mesma. Devido a que cada partícula possui uma velocidade, o comportamento do PSO é direcional, isto é, a partícula é acelerada estocasticamente na direção da melhor posição global da população e também é acelerada estocasticamente na direção de sua anterior melhor posição. C_1 e C_2 são conhecidos como os parâmetros cognitivo e social, respectivamente. ω é o fator de inércia (ou fator de amortecimento) que controla o impacto da velocidade previa da partícula sobre sua velocidade atual (SHI; EBERHART, 1998). O fator de inércia no PSO lembra o parâmetro de temperatura T no algoritmo *Simulated Annealing*. Um fator de inércia alto facilita uma exploração global do espaço de busca, enquanto que um valor pequeno deste parâmetro possibilita uma

busca local (ajuste fino), portanto, a apropriada seleção do fator de amortecimento fornece um balance entre a capacidade de busca local e de busca global do algoritmo e permitirá um número menor de iterações (em media) para identificar o ponto ótimo global. A Equação (51) possibilita transições aleatórias nas posições das partículas, permitindo que o algoritmo escape dos pontos ótimos locais.

O algoritmo *PSO* funciona mediante a modificação da distância que cada partícula percorre em cada direção (Equação (50)). Para controlar o passo do algoritmo (v_{k+1}^i) e para prevenir o fenômeno de explosão (TAYAL, 2003), um valor de velocidade máxima $Vmax = (pUB-pLB0) / H$ (com $H = 5$) foi empregado nesta pesquisa. Vale a pena salientar que todas as posições das partículas p_k^i devem ser limitadas por seus valores máximo (pUP) e mínimo (pLB) permitidos.

A literatura propõe usar $0 < \omega < 1.4$, $C_1 = C_2 = 2$ com $C_1 + C_2 \leq 4$ e $5 < H < 10$ para manter um equilíbrio entre a capacidade de busca global e local do algoritmo (PARSOPOULOS; VRAHATIS, 2002; KENNEDY; EBERHART, 1995; KENNEDY, 1997; ALRASHIDI; EL-HAWARY, 2006; TAYAL, 2003). Como sabido, os três parâmetros ω , C_1 e C_2 , são dependentes do problema estudado (VENTER; SOBIESZCZANSKI-SOBIESKI, 2003; SCHUTTE; GROENWOLD, 2003). Devido a este fato, inclusive usuários experientes do *PSO* devem realizar testes exaustivos para encontrar o melhor conjunto de parâmetros para o *PSO* e usuários menos experientes podem fornecer valores inadequados causando a falha do algoritmo.

4.3.2 Topologias de Vizinhança

No PSO, as partículas são influenciadas pelo sucesso de qualquer outra partícula com a qual elas estão conectadas. Essa vizinhança, não é constituída necessariamente por partículas que estão próximas umas das outras (em termos do valor das variáveis), mas de partículas que estão próximas em função da topologia de vizinhança que define a estrutura social do enxame. Neste sentido, a vizinhança determina o conjunto de partículas que contribuem para o cálculo do b_k^g de uma partícula dada.

Dependendo da vizinhança utilizada, o PSO pode ter duas versões. Se b_k^g , na Equação (51), for a melhor partícula do enxame (a melhor posição de toda a população), tem-se a versão global do PSO, ilustrada na Figura 5 e chamada de topologia totalmente conectada. Por outro lado, a versão local se obtém substituindo b_k^g pela melhor posição dentro de um número de partículas p ($p < N$, onde N é o número total de partículas da população) vizinhas da partícula p_k^i . Existem varias opções para definir uma vizinhança local, as mais usadas são descritas a seguir.

Vizinhança Local: Nesta topologia, cada partícula é afetada pela melhor partícula de suas p vizinhas imediatas. Geralmente, o valor $p = 2$ é usado, dando origem à topologia conhecida como anel, mostrada na Figura 5. Neste caso, $b_k^g =$ melhor vizinha imediata, na Equação (51).

Vizinhança Focal: uma partícula esta conectada com todas as outras e elas estão conectadas unicamente com a primeira (chamada de partícula

focal) como ilustrado na Figura 5. Neste caso, b_k^g = partícula foco, na equação (51).

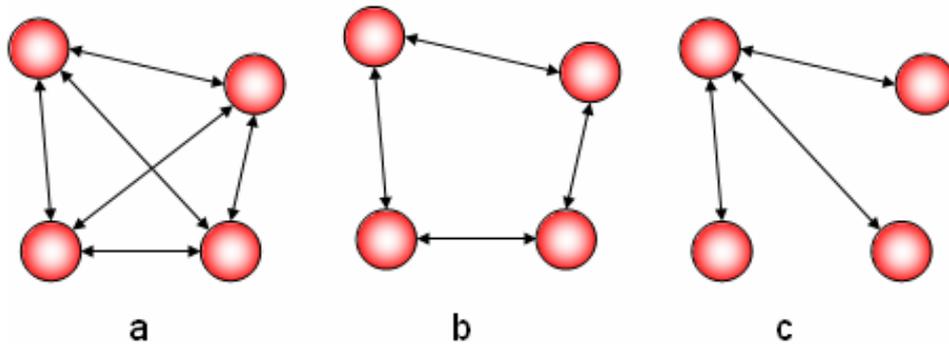


Figura 5. Topologias de vizinhança no PSO. a) totalmente conectada. b) local e c) focal.

A topologia de vizinhança tem influencia na taxa de convergência do PSO, já que, determina quanto tempo empregaram as partículas para localizar a melhor posição no espaço de busca. Por exemplo, na versão global, todas as partículas estão conectadas e recebem informação sobre a melhor solução ao mesmo tempo. Desta forma, usando esta estrutura de vizinhança global, o enxame tende a convergir com maior rapidez que quando são usadas topologias locais o de estrela, devido a que, no último caso, a informação sobre a melhor posição demora mais em ser transmitida. Porém, pela mesma razão, a versão global é mais suscetível de convergir de forma prematura para um mínimo local.

4.3.3 Critérios de Parada

Um critério de parada robusto é importante para qualquer algoritmo de otimização para evitar avaliações de função objetivo adicionais depois que

uma solução ótima foi encontrada. Idealmente, o critério de parada empregado não deve ter nenhum parâmetro relacionado com o problema. O critério de convergência considerado neste trabalho é básico. A máxima variação na função objetivo foi monitorada para um número específico de iterações consecutivas. Se a variação máxima da função objetivo foi menor que uma variação predefinida, se assume a convergência do algoritmo.

A listagem do programa PSO em FORTRAN utilizado neste estudo é apresentada no Apêndice B e o pseudo código na Figura 6.

```

k=0
Iniciar PSO:
Iniciar aleatoriamente N partículas  $p_0^i$ 
Iniciar aleatoriamente as velocidades iniciais  $v_0^i$ 
Avaliar a função objetivo  $fobj(p_0^i)$  para cada partícula
Calcular  $b_0^i$  e  $b_0^g$ 
While (k < 80)
  Calcular nova velocidade  $v_{k+1}^i$  usando vetor p (Equação 51)
  IF (  $v_{k+1}^i > Vmax$  or  $v_{k+1}^i < -Vmax$  ) then fazer  $v_{k+1}^i = Vmax$  ou  $v_{k+1}^i = -Vmax$ 
  END IF
  Calcular nova posição  $p_{k+1}^i$  (Equação 50)
  IF (  $p_{k+1}^i < pLB$  or  $p_{k+1}^i > pUB$  ) then fazer  $p_{k+1}^i = pLB$  ou  $p_{k+1}^i = pUB$ 
  END IF
  Avaliar a função objetivo =  $f_{obj}(p_{k+1}^i)$  para cada partícula
  Atualizar  $b_{k+1}^i$  e  $b_{k+1}^g$ 
  IF tolerância < 1E-6 then  $f(X) = fobj(b_{k+1}^g) \rightarrow$  end PSO.
  Else k=k+1
  w = w*0.98
END IF
END

```

Figura 6. Pseudocódigo do PSO

4.4 Algoritmos Genéticos (AGs)

OS Algoritmos Genéticos podem ser vistos como um processo evolucionário de otimização onde uma população de soluções evolui através de uma seqüência de gerações. Durante cada geração, o valor da função objetivo (*fitness*) é calculado, e as soluções são selecionadas segundo este

valor. A probabilidade de sobrevivência de uma solução é proporcional a seu valor de *fitness*. Este processo se baseia no princípio de sobrevivência do mais adaptado. As soluções obtidas são recombinadas, através dos processos de mutação e cruzamento, abordados na seguinte seção.

4.4.1 Operadores e Parâmetros Básicos dos AGs

Um algoritmo genético simples emprega três operadores básicos: reprodução, cruzamento e mutação. A reprodução é um processo no qual os indivíduos (soluções) são copiados de acordo com seu valor da função objetivo. O cruzamento requer da união de pelo menos dois indivíduos selecionados de forma aleatória. A informação destes indivíduos é parcialmente compartilhada de acordo com um ponto de cruzamento selecionado de forma randômica. O cruzamento se aplica para transmitir informação importante dos pais para os filhos e é aplicado com certa probabilidade. A mutação é uma alteração ocasional do valor de uma variável (indivíduo) e ajuda ao algoritmo a fugir de pontos ótimos locais.

Os efeitos do cruzamento, geralmente variam durante uma rodada. No começo, a população é aleatória de maneira que o cruzamento tem efeitos significativos, deslocando indivíduos grandes distancia no espaço de busca. No final da rodada, a população convergiu, isso significa que os indivíduos têm valores similares (*fitness*) e por esta razão, o cruzamento tem um efeito relativamente pequeno. Adicionalmente, a probabilidade de cruzamento é, algumas vezes, modificada durante a rodada, começando com valores altos e terminando com valores muito pequenos, se acrescentado de esta forma, um grau mais de complexidade.

O efeito da mutação tende a ser oposto ao do cruzamento. Na fase inicial a mutação tem menos influencia sobre a população e esta aumenta nas etapas finais. Isto se deve a que, a população inicial é aleatória, fazendo com que qualquer variação inicial nos indivíduos não cause uma mudança tão dramática. Já no final da rodada, quando a população tem convergido, as variações podem ser significativas. Se a probabilidade de mutação muda durante uma rodada, é comum começar com um valor pequeno e aumentá-lo perto do final.

A função da seleção, no algoritmo genético, é garantir a sobrevivência do mais adaptado, um conceito central dos algoritmos evolucionários. A seleção pode ser implementada de varias formas, incluindo mecanismos como o torneio e a roleta. Na roleta a probabilidade de seleção é proporcional ao valor da função objetivo (*fitness*) do individuo. No torneio são escolhidos aleatoriamente dois indivíduos da população e aquele com melhor *fitness* é selecionado.

Da exposição anterior, fica evidente que os valores dos parâmetros de mutação (probabilidade de mutação (P_m)), cruzamento (probabilidade de cruzamento (P_c)) e seleção (estratégia de seleção dos pais) influenciam fortemente o desempenho global do algoritmo. Adicionalmente, existem outras constantes que o usuário deve definir antes de utilizar com sucesso um algoritmo genético. Esses parâmetros são listados a seguir

- tamanho da população;
- a distribuição de probabilidade para gerar a população inicial;
- número máximo de iterações (N_i).

A Figura 7 apresenta o fluxograma de um algoritmo genético básico.

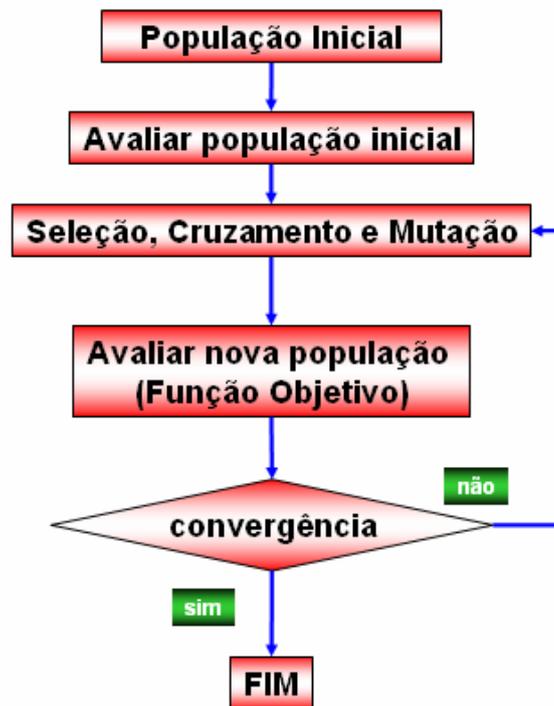


Figura 7. Fluxograma algoritmo genético básico.

4.4.2 Critérios de Parada

O algoritmo genético pára quando o valor da função objetivo, de pelo menos um individuo da população, atinge um valor de tolerância predefinido ou quando o número máximo de gerações (iterações) estabelecido é alcançado.

5 O novo algoritmo híbrido: PSOS (PSO-Simplex)

5.1 Introdução

De forma geral, a literatura clássica sobre otimização aborda algoritmos para calcular soluções de problemas de otimização com restrições. A maior parte dos métodos tradicionais para PNL consiste de estratégias de busca local, que tem a séria desvantagem de convergirem para pontos ótimos locais. Técnicas como a família de métodos de Newton-Gauss utiliza gradientes e/ou hessianas para determinar o mínimo de funções quadráticas num passo só. Estes procedimentos são efetivos quando a informação sobre derivadas está disponível, o qual não é comum na maioria das situações reais.

Para contornar os anteriores problemas, diversos algoritmos baseados em heurísticas vêm sendo empregados com sucesso. Neste contexto, é necessário fazer algumas definições. O termo **Algoritmo Exato** (ou *Clássico*) se refere a um procedimento que calcula, de forma demonstrável num tempo determinado, uma solução ótima (por exemplo, as técnicas de programação dinâmica). Um **algoritmo heurístico** (o simplesmente, uma **heurística**) é um algoritmo que não tem garantia de encontrar uma solução ótima, porém,

tem a capacidade de encontrar soluções quase ótimas ou ótimas de forma rápida quando comparadas com uma busca exaustiva. A heurística pode ser **construtiva** (produzir uma única solução) ou pode ser uma **busca local** (começar de uma ou mais soluções aleatórias e procurar, de forma iterativa, soluções na vizinhança dos pontos iniciais) ou uma combinação das duas anteriores. O novo PSOS pode ser classificado como um **algoritmo heurístico construtivo**.

A aplicação direta de um método heurístico (por exemplo, os AGs, o PSO ou o SA) para a solução do problema DD não necessariamente funciona, particularmente com relação à precisão, à estabilidade e ao custo computacional requeridos para obter uma solução ótima. Adicionalmente, um dos pontos críticos no uso destes métodos é a seleção de um grande número de parâmetros que controla seu desempenho (PARSOPOULOS; VRAHATIS, 2002; SU et.al., 2005; MOITA et.al., 2006). Por estes motivos, a busca por algoritmos autoconfigurados se faz necessária.

5.2 Algoritmos Autoconfigurados

A implementação de um algoritmo heurístico é definida não só pela forma de representação do problema, da estrutura de dados, mas também por uma configuração, isto é, certo conjunto de valores dos parâmetros do algoritmo, como por exemplo, o fator de amortecimento e as constantes C_1 e C_2 no PSO, a taxa de esfriamento e a temperatura inicial no SA ou a taxa de mutação e de cruzamento no AG, que devem ser definidos de forma completa para conseguir rodar com sucesso o código computacional. Infelizmente, a definição desses parâmetros é feita, na maior parte da literatura, depois de

cuidadosos e dispendiosos ajustes. Como resultado, o melhor desempenho do algoritmo tem significado só com relação à configuração particular considerada para os problemas estudados, impossibilitando-se assim sua aplicação automática e independente, em outros tipos de problemas. Neste contexto, um dos objetivos do algoritmo híbrido, proposto nesta pesquisa, é incorporar uma estratégia de busca de uma configuração ótima ou quase ótima para o PSO, através do método Simplex não linear (NELDER; MEAD, 1965), que o torne independente do tipo de problema estudado. Este algoritmo é chamado aqui de autoconfigurado. O algoritmo Simplex é introduzido na seguinte seção com a finalidade de facilitar o entendimento do PSOS, apresentado na parte final do capítulo.

5.3. O Algoritmo Nelder - Mead (N-M)

O Simplex é um método de busca direta muito empregado na área de programação não linear. Neste trabalho, foi utilizada a versão original do popular algoritmo Simplex (NELDER; MEAD, 1965), onde $n+1$ pontos (esses pontos pertencem a R^n) são utilizados para construir o Simplex inicial. A seguir se descreve uma iteração típica do algoritmo.

Em cada iteração, o pior ponto (X_w) no Simplex é achado ordenando seus vértices da seguinte forma: $X_1, X_2, \dots, X_n, X_{n+1}$, de maneira que $f(X_1) < f(X_2) < \dots < f(X_n) < f(X_{n+1})$. Em um problema de minimização, X_1 é o melhor ponto, X_{n+1} é o pior ponto, X_n é o segundo pior ponto e assim por diante. Primeiramente, o centróide (X_c) de todos, menos o pior ponto, é calculado. Seguidamente, o pior ponto do Simplex é refletido (ver Equação (52)) com relação ao centróide e o novo ponto (X_r) é determinado. Se $f(X_1) \leq f(X_r) < f(X_n)$,

e se as restrições não são violadas neste ponto, se considera que a reflexão levou o Simplex a uma melhor região no espaço de busca, o pior ponto X_{n+1} é substituído por X_r e a iteração acaba.

Por outro lado, se $f(X_r) < f(X_1)$, um ponto de expansão (X_e) (Equação (53)) ao longo da direção desde o centróide até o ponto refletido é calculado. Se $f(X_e) < f(X_r)$, X_e é aceito ($X_{n+1} = X_e$) e a interação acaba. Se $f(X_e) \geq f(X_r)$, então X_r é aceito ($X_{n+1} = X_r$) e a iteração termina. Se $f(X_r) \geq f(X_n)$, existem duas possibilidades nas quais a contração na direção do centróide é feita: primeira, se $f(X_r) \geq f(X_{n+1})$ e segunda, se $f(X_n) \leq f(X_r) < f(X_{n+1})$. Para a primeira possibilidade, o ponto de contração é calculado usando a Equação (54). Se $f(X_{cont1}) < f(X_{n+1})$, X_{cont1} é aceito ($X_{n+1} = X_{cont1}$) e a iteração finaliza, do contrário uma redução é realizada (ver Equação (56)). Para a segunda opção, o ponto de contração é calculado usando a Equação (55). Se $f(X_{cont2}) \leq f(X_r)$, X_{cont2} é aceito ($X_{n+1} = X_{cont2}$), e a iteração termina, de outra forma uma redução é realizada. Quando uma redução acontece, a função é avaliada nos n novos pontos P_i dados pela Equação (56). Esses pontos (X_1, P_2, \dots, P_{n+1}) são os novos vértices do simplex na seguinte iteração. O procedimento se repete até que o seguinte critério de convergência é satisfeito: se o desvio padrão dos valores da função nos $n+1$ pontos atuais é menor que uma tolerância dada, o algoritmo para.

$$X_r = X_c + \rho(X_c - X_{n+1}) \quad (52)$$

$$X_e = X_c + \chi(X_r - X_c) \quad (53)$$

$$X_{cont1} = X_c - \gamma(X_c - X_{n+1}) \quad (54)$$

$$X_{cont2} = X_c + \gamma(X_r - X_c) \quad (55)$$

$$P_i = X_1 + \sigma(X_i - X_1), \quad i = 2, \dots, n+1. \quad (56)$$

A Figura 8 ilustra três iterações típicas do Simplex na função de Himmelblau em duas dimensões. Através de sucessivas aplicações das Equações (52) a (56) o método converge para o ponto ótimo global.

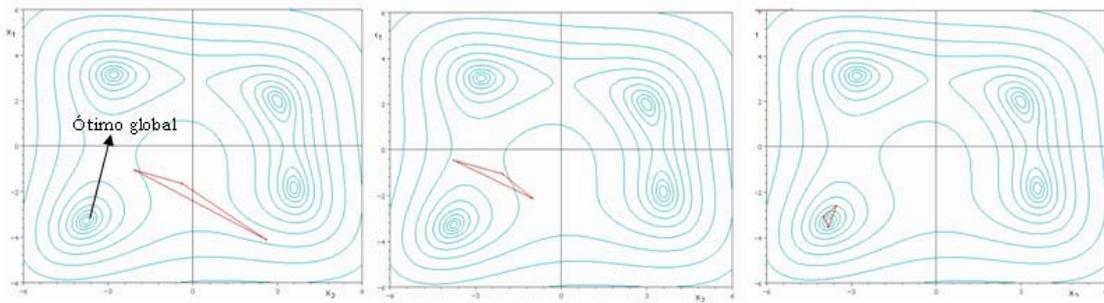


Figura 8. Dinâmica do método Simplex.
(http://allrss.com/wikipedia.php?title=Image:Nelder_Mead2.gif).

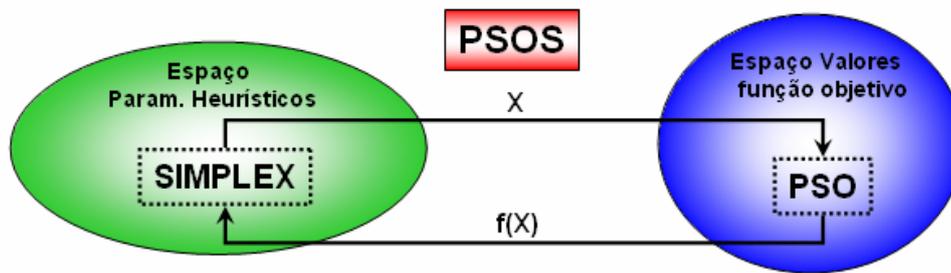
5.4 O Algoritmo PSOS

A nova estratégia que se propõe neste trabalho, para controlar a seleção adequada das constantes heurísticas que governam o comportamento do PSO, se baseia no uso do método Simplex (NELDER; MEAD, 1965). Apesar das conhecidas falhas e ineficiências do algoritmo Simplex (LAGARIAS et. al., 1998) ele é usado aqui para escolher os parâmetros heurísticos do PSO e, desta forma, fazer com que o PSO se torne independente desses valores, como explicado a seguir.

5.4.1 Heurística do PSOS

A idéia central do método é que o algoritmo Simplex selecione os valores dos parâmetros N , w , C_1 e C_2 contidos no espaço de parâmetros (ver Figura 9) e, assim, procure uma configuração ótima ou quase ótima para o PSO.

Dentro desta heurística, cada vértice do Simplex fica definido pelas coordenadas (N, w, C_1, C_2) . Nesta situação, o PSO toma os valores dos parâmetros heurísticos determinados pelo Simplex e avalia a função objetivo do problema (ver Figura 9). Conseqüentemente, cada ponto do Simplex, e qualquer reflexão, contração, expansão e redução (ver seção 5.3), é avaliada com um enxame independente, caracterizado pelo vetor $X_i (N_i, w_i, C_{1i}, C_{2i})$ $i=1, \dots, n+1$, onde, n é o número de parâmetros do PSO e N_i é um número inteiro, que representa o tamanho do enxame. A variável N_i (tamanho do enxame) foi incluída com a finalidade de estimar seu valor ótimo. Este procedimento melhora a capacidade de busca do algoritmo híbrido e o torna independente dos parâmetros heurísticos, devido a que realiza uma busca, dirigida por parâmetros quase ótimos, de forma automática.



a)

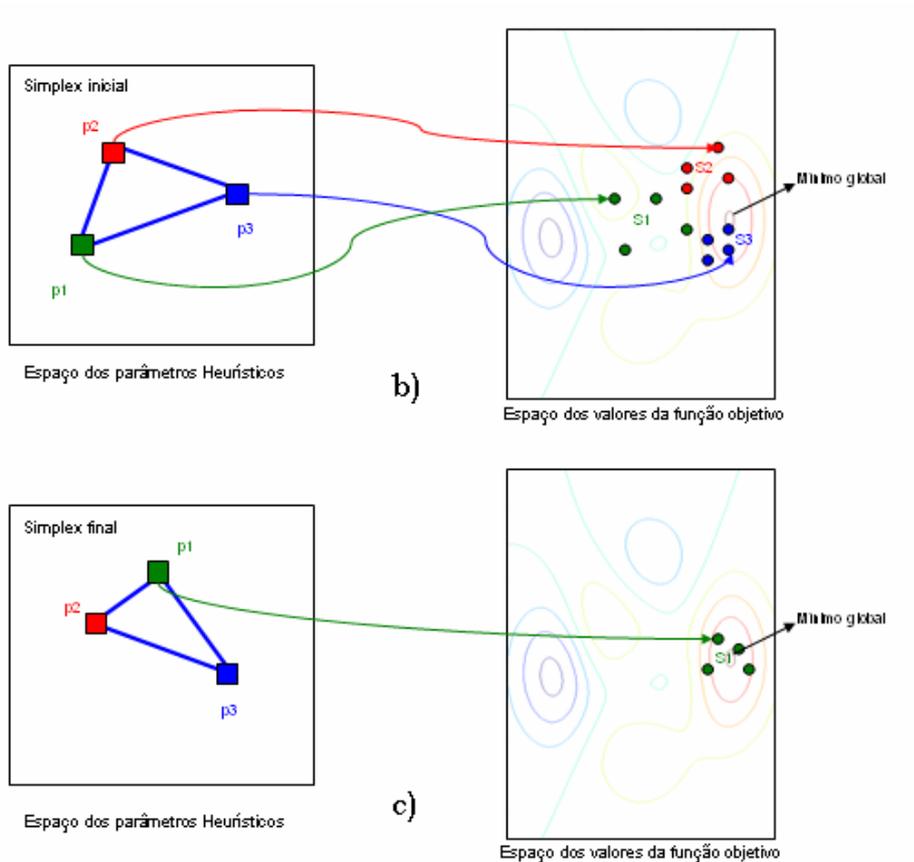


Figura 9. Heurística do PSOS. a) Funcionamento global. b) Iteração inicial (Simplex em duas dimensões) e c) Iteração final após uma reflexão.

O pseudocódigo para o PSOS desenvolvido para esta pesquisa é apresentado a seguir.

Iniciar o algoritmo PSOS.

Definir ponto inicial X_1 : $N_1 \leftarrow a$, $w_1 \leftarrow b$, $C_{11} \leftarrow c$ $C_{21} \leftarrow d$ (a, b, c, d tentativa inicial) do Simplex

Gerar aleatoriamente $n+1$ vértices do Simplex (X_i) $i = 2, \dots, n+1$

Call PSO ($X, f(X)$) para avaliar a função objetivo em: X_1, \dots, X_{n+1}

$j = 0$

WHILE ($j < \text{termination criterion1}$)

Ordenar os vértices X_1, \dots, X_{n+1} de forma que: $f(X_1) <, \dots, < f(X_{n+1})$.

Refletir o pior ponto: calcular X_r (Equação (52))

Call PSO ($X, f(X)$) para avaliar $f(X_r)$

IF $f(X_1) \leq f(X_r) < f(X_n) \rightarrow (X_{n+1} = X_r)$.

ELSE

IF $f(X_r) < f(X_1) \rightarrow$ calcular um ponto de expansão X_e (Equação (53)).

Call PSO ($X, f(X)$) para calcular $f(X_e)$

If $f(X_e) < f(X_r) \rightarrow (X_{n+1} = X_e)$

else

$(X_{n+1} = X_r)$

End if

ELSE

If $f(X_r) \geq f(X_n)$

If $f(X_n) \leq f(X_r) < f(X_{n+1}) \rightarrow$ avaliar X_{cont2} (Equação (55))

Call PSO ($X, f(X)$) para avaliar $f(X_{cont2})$

If $f(X_{cont2}) \leq f(X_r) \rightarrow (X_{n+1} = X_{cont2})$

Else realizar redução (Equação (56)).

End if

Else

If $f(X_r) \geq f(X_{n+1}) \rightarrow$ Avaliar X_{cont1} (Equação (54)).

Call PSO ($X, f(X)$) para calcular $f(X_{cont1})$

If $f(X_{cont1}) < f(X_{n+1}) \rightarrow (X_{n+1} = X_{cont1})$

Else realizar redução (Equação (56)).

End if

Else

$(X_{n+1} = X_r)$

End If

End if

Else

$(X_{n+1} = X_r)$

End if

END IF

END IF

If convergence criterion1 é satisfeito \rightarrow End Simplex.

Else $j=j+1$.

End if

END

SUBROUTINE PSO (X, f(X))

$k=0$

Iniciar PSO:

Iniciar aleatoriamente N partículas p_0^i

Iniciar aleatoriamente as velocidades iniciais v_0^i

Avaliar a função objetivo $f_{obj}(p_0^i)$ para cada partícula

Calcular b_0^i e b_0^g (como descrito anteriormente)

While ($k <$ termination criterion2)

Calcular nova velocidade v_{k+1}^i usando vector X (Equação (51))

If ($v_{k+1}^i > Vmax$ or $v_{k+1}^i < -Vmax$) then fazer $v_{k+1}^i = Vmax$ ou $v_{k+1}^i = -Vmax$

End if

Calcular nova posição p_{k+1}^i (Equação (50))

If ($pLB > v_{k+1}^i$ or $v_{k+1}^i > pUB$) then fazer $p_{k+1}^i = pLB$ ou $p_{k+1}^i = pUB$

End if

Avaliar a função objetivo = $f_{obj}(p_{k+1}^i)$ para cada partícula

Atualizar b_{k+1}^i e p_{k+1}^g

If convergence criterion2 é satisfeito then $f(X) = f_{obj}(p_{k+1}^g) \rightarrow$ end

PSO.

Else $k=k+1$

$w = w*0.98$

End if

END

O fluxograma do PSO é apresentado na Figura 10.

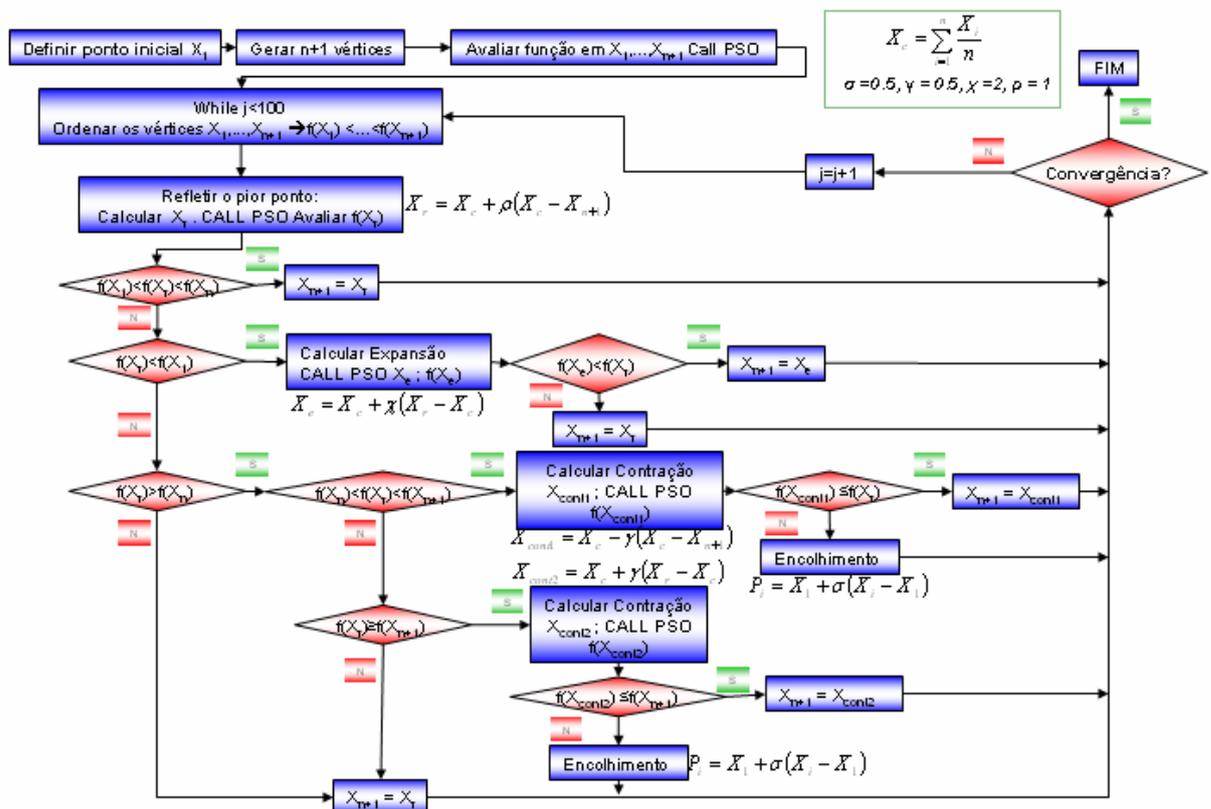


Figura 10. Fluxograma do algoritmo PSOS

Experimentos paralelos tentando combinar o PSO e o SA, de forma tal que o PSO controlasse a seleção dos parâmetros do SA e vice-versa, forneceram resultados pouco promissores, devido ao alto custo em termos de número de avaliações da função objetivo e a sua falta de estabilidade. Por esta razão a pesquisa nesse sentido foi abandonada.

5.4.2 Critérios de Parada do PSOS

No algoritmo descrito acima, a redução linear de inércia foi usada dentro da sub-rotina PSO. O controle do limite de velocidade está implicitamente incluído no algoritmo (*Vmax*). O máximo número de iterações permitido para o PSO (*termination criterion2*) foi de 80 e de 100 (*termination criterion1*), para o Simplex. Para a sub-rotina PSO, os últimos dez valores da

função foram monitorados. Quando seu desvio padrão foi menor que $1E-6$ (*stopping criterion*2), a convergência foi assumida e o novo ponto foi enviado para o Simplex e uma nova iteração do Simplex começou. A tolerância permitida para o laço do Simplex foi $1E-6$. Os coeficientes utilizados para o método Nelder-Mead foram $\sigma = 0.5$, $\gamma = 0.5$, $\alpha = 2$, $\rho = 1$ (caso padrão).

5.5 Análise de Convergência

As bases teóricas para análises de convergência dos algoritmos heurísticos ainda são fracas em comparação com os resultados experimentais; em outras palavras, sabe-se que eles funcionam, porém, não se sabe exatamente porque.

Felizmente, alguns resultados teóricos sobre convergência têm sido reportados. Por exemplo, para o SA, Laarhoven e Aarts (1987), produziram o primeiro resultado teórico sobre sua convergência. Locatelli (2000), provou a convergência do SA no caso de problemas de otimização com variáveis contínuas. Com relação ao PSO, recentemente Kadiramanathan, Selvarajah e Fleming (2006), publicaram resultados sobre sua convergência e estabilidade.

Nesta pesquisa, é apresentada evidencia experimental sobre a convergência e estabilidade do algoritmo PSOS. Espera-se que uma explicação teórica destas propriedades siga o caminho aberto por Kadiramanathan, Selvarajah e Fleming (2006).

O grande número de publicações na área de heurísticas vem apresentado um aumento estável, quase exponencial, isto mostra que os pesquisadores continuam tentando projetar melhores algoritmos para ser

usados em problemas práticos. Talvez, o sucesso experimental das técnicas desenvolvidas tenha feito com que as análises teóricas passem a segundo plano.

6 Avaliação Experimental dos Algoritmos

6.1 Introdução

O potencial e as limitações do algoritmo apresentado neste estudo são demonstrados mediante sua aplicação em funções teste (exemplos matemáticos de otimização) que tem aparecido recentemente na literatura especializada. A efetividade do PSOS foi avaliada mediante comparação com o algoritmo SA proposto por Corana (1987), com um Algoritmo Genético desenvolvido por Goldberg, (1989) e com o PSO.

6.2 Critérios de Avaliação dos Algoritmos: Precisão, Estabilidade, Robustez, Custo Computacional e Confiança

Levando-se em conta que os algoritmos estudados neste trabalho são de natureza estocástica, os critérios para avaliar seu desempenho são descritos a seguir. O desvio padrão e a distância entre a media e o valor ótimo analítico das funções estudadas foram usados para medir a Precisão e a Estabilidade dos métodos. Um método de otimização heurístico é dito

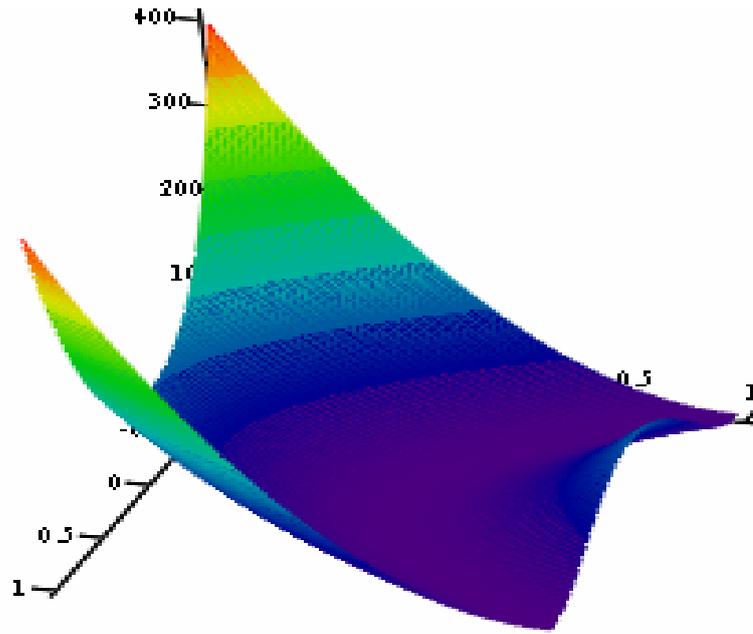
Estável, se seu desvio padrão é baixo. O método tem Precisão se, além de cumprir a condição anterior, a distância entre a média de m rodadas e o valor ótimo analítico é pequena. O algoritmo pode ser qualificado como Robusto se, quando testado em problemas diferentes, ele apresenta Precisão e Estabilidade.

A Confiança C do algoritmo é definida como o número de vezes de um total de m testes que o algoritmo encontra um valor da função dentro de uma precisão especificada.

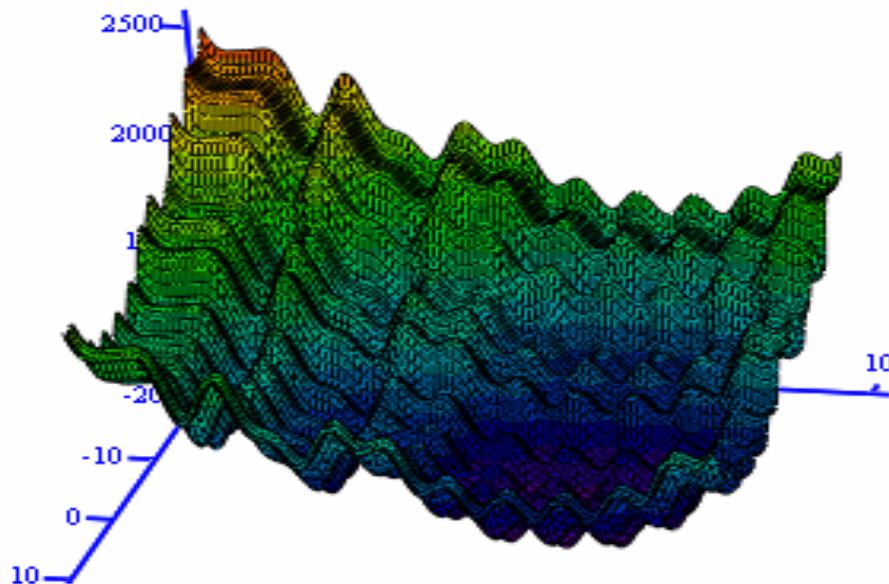
Finalmente, o Custo Computacional dos algoritmos foi medido em termos do tempo total de computação e do número total de avaliações da função objetivo para obter a solução.

6.3 Funções Teste - Resultados

O primeiro teste foi feito na função de Rosenbrock em duas dimensões (R2D). Esta função é um vale em curva cujo fundo desce com uma declividade muito suave e tem seu mínimo global no ponto (1, 1), com um valor da função igual a zero (Figura 11). O segundo teste foi conduzido na função de Brown em 20 dimensões (B20D), a qual tem um mínimo global no ponto (0, 0) e o valor da função nesse ponto igual a zero. O terceiro teste foi realizado na função apresentada por Venter e Sobieszczanski-Sobieski (VS) (2003). Esta função possui uma considerável quantidade de mínimos locais e seu valor mínimo global tem um valor de 1000 no ponto (0, 0) como indicado na Figura 11.



a)



b)

Figura 11. Funções teste em duas dimensões. a) Função de Rosenbrock e b) Função de Venter e Sobieszczanski-Sobieski.

As anteriores funções constituem testes difíceis para qualquer procedimento de otimização. Os domínios admissíveis das funções foram:

$[-1000 \leq X_i \leq 1000]$ para a função R2D, $[-1 \leq X_i \leq 4]$ para a função B20D e $[-10 \leq X_i \leq 50]$ para a função VS.

Os parâmetros básicos do PSO empregados para os exemplos foram $w=1.4$ (com um fator de redução de inércia igual a 0,98), $C_1=C_2=2$ e $H=5$. Este conjunto de valores é empregado frequentemente nos estudos reportados na literatura especializada. Para o número de partículas empregadas, dois casos foram analisados: N igual ao valor médio de número de partículas calculado pelo PSOS +/- o desvio padrão do número de partículas ($N \approx MV \pm SD$, ver tabelas 2, 3 e 4), permitindo, desta forma, realizar uma comparação entre o desempenho dos algoritmos. O PSO básico realizou um número máximo de avaliações da função objetivo dado pela multiplicação do número de iterações permitidas (80) vezes o número de partículas utilizadas. Na coluna nomeada como PSO*, o número máximo de avaliações permitidas foi de um milhão. Os resultados empregando estes valores são apresentados nas tabelas 2, 3 e 4.

Em todos os testes, o enxame inicial (*swarm*) foi distribuído aleatoriamente e os valores iniciais do vértice do Simplex foram gerados com ajuda de um gerador de números aleatórios (JAFPE, 2000). O algoritmo SA também foi iniciado em diferentes pontos, escolhidos de forma randômica e com diferentes sementes. As temperaturas iniciais foram 50000 para a função R2D, 50 para a VS e 0,5 para a B20D. Finalmente, o número máximo de iterações permitidas foi de quatro milhões. Os últimos quatro valores finais da função foram usados para decidir a parada do algoritmo SA e uma tolerância de 1E-6 foi usada como recomendado por Corana et.al., (1987) e Goffe, Ferrier e Rogers, (1994).

No algoritmo genético usado, os parâmetros empregados são definidos a seguir. Foi selecionada uma população de 500 indivíduos, distribuídos de forma aleatória no espaço do problema. O número máximo de iterações permitidas foi de 10000. A probabilidade de mutação P_m foi definida como 0,01 e a probabilidade de cruzamento P_c foi de 0,5. O mecanismo de seleção escolhido foi o da roleta. Adicionalmente, no AG utilizado, foi implementado o processo conhecido como elitismo, no qual, o melhor indivíduo de uma geração é passado para a seguinte sem modificações. O elitismo busca melhorar a velocidade de busca do algoritmo.

Cada rodada dos algoritmos foi repetida mil vezes e o melhor valor da função (MF), o pior valor da função (PF), a média dos valores da função (MEF), o desvio padrão dos valores da função (DF), o melhor número de chamadas da função (MC), o pior número de chamadas da função (PC), a média do número de chamadas da função (MEC), o desvio padrão do número de chamadas da função (DC) e a confiança C (o número de vezes de um total de mil que o algoritmo encontrou um valor da função dentro de uma precisão $1E-4$ do valor ótimo analítico) são reportados.

Tabela 2. Comparação entre o *Simulated Annealing* (SA), o PSO básico, o PSOS e o AG na função de Rosenbrock.

$$\text{R2D: } f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

| Estatística | PSOS | SAA | PSO 150 | PSO 320 | PSO* | AG |
|--------------------|---------|---------|---------|---------|--------|-------|
| MF | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| PF | 0.0 | 1065,27 | 943,51 | 848,11 | 872.04 | 1.0 |
| MEF | 0.0 | 29,74 | 32,90 | 30,96 | 18.98 | 0.39 |
| DF | 1.12E-7 | 1.47E2 | 1.11E2 | 1.05E2 | 0.79E2 | 0.53 |
| MC | 9.52E4 | 4.17E5 | 9.9E3 | 9.75E3 | 8.85E3 | 2.1E4 |
| PC | 6.041E5 | 9.42E5 | 1.2E4 | 1.2E4 | 3.82E4 | 1.5E5 |
| MEC | 3.52E5 | 4.75E5 | 1.194E4 | 1.195E4 | 1.7E4 | 9.4E4 |
| DC | 1.35E5 | 1.99E5 | 2.25E2 | 2.10E2 | 4.9E3 | 4.7E5 |
| t _m (s) | 45 | 68 | 21 | 28 | 33 | 37 |
| C | 100,0% | 95,0% | 41,0% | 80,0% | 49,0% | 40,0% |

melhor valor da função (MF); pior valor da função (PF); média dos valores da função (MEF); desvio padrão dos valores da função (DF); melhor número de chamadas da função (MC); pior número de chamadas da função (PC); média do número de chamadas da função (MEC); desvio padrão do número de chamadas da função (DC); confiança (C); tempo médio de cálculo (t_m).

Tabela 3. Comparação entre o *Simulated annealing* (SA), o PSO básico, o PSOS e o AG na função de Brown.

$$\text{Brown 20D: } f(x_i) = \sum_{i=1}^{19} (x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)}$$

| Estatística | PSOS | SAA | PSO 217 | PSO 331 | PSO* | AG |
|--------------------|---------|---------|---------|---------|--------|--------|
| MF | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| PF | 0.0 | 0.0 | 4.0 | 4.0 | 5.0 | 4.5 |
| MEF | 0.0 | 0.0 | 0,4115 | 0,2339 | 1.8 | 0.95 |
| DF | 1,76E-5 | 6,43E-8 | 8,35E-1 | 6,28E-1 | 0.22E1 | 0.78E1 |
| MC | 3.6E5 | 2.76E6 | 1.73E4 | 264E4 | 4.03E4 | 8.5E5 |
| PC | 1.008E6 | 2.97E6 | 1.73E4 | 264E4 | 5.19E4 | 1.5E6 |
| MEC | 5.23E5 | 2.88E6 | 1.73E4 | 264E4 | 4.59E4 | 1.2E5 |
| DC | 1.52E5 | 3.27E4 | 0,0000 | 0,0000 | 4.08E3 | 4.6E5 |
| t _m (s) | 53 | 96 | 35 | 33 | 49 | 58 |
| C | 100,0% | 100,0% | 8,0% | 51,0% | 60,0% | 60% |

melhor valor da função (MF); pior valor da função (PF); média dos valores da função (MEF); desvio padrão dos valores da função (DF); melhor número de chamadas da função (MC); pior número de chamadas da função (PC); média do número de chamadas da função (MEC); desvio padrão do número de chamadas da função (DC); confiança (C); tempo médio de cálculo (t_m).

Tabela 4. Comparação entre o *Simulated annealing* (SA), o PSO básico, o PSOS e o AG na função de Venter.

| Função VS: | | | | |
|--|---------|----------|-------------|---------|
| $f(x_1, x_2) = x_1^2 - 100\cos(x_1)^2 - 100\cos(x_1^2/30) + x_2^2 - 100\cos(x_2)^2 - 100\cos(x_2^2/30) + 1400$ | | | | |
| Estatística | PSOS | SAA | PSO 20 Part | AG |
| MF | 1000.00 | 1000.00 | 1000.00 | 1000.00 |
| PF | 1000.00 | 1000.00 | 1017.63 | 1000.46 |
| MEF | 1000.00 | 1000.00 | 1000.81 | 1000.19 |
| DF | 2.73E-4 | 8.99E-11 | 0.34E1 | 0.23 |
| MC | 4E3 | 4.04E5 | 7.5E2 | 3.45E4 |
| PC | 3.7E4 | 4.84E5 | 1.2E3 | 1.48E5 |
| MEC | 1.73E4 | 4.49E5 | 1059.64 | 1.20E5 |
| DC | 1.14E4 | 4.25E4 | 105.83 | 4.91E4 |
| t_m (s) | 12 | 36 | 6 | 8 |
| C | 100.0% | 100.0% | 91.0% | 40.0% |

melhor valor da função (MF); pior valor da função (PF); média dos valores da função (MEF); desvio padrão dos valores da função (DF); melhor número de chamadas da função (MC); pior número de chamadas da função (PC); média do número de chamadas da função (MEC); desvio padrão do número de chamadas da função (DC); confiança (C); tempo médio de cálculo (t_m).

Nas funções VS, R2D e B20D, o PSO básico (ver Tabelas 2, 3 e 4) não achou facilmente o ótimo global e em muitos casos, nunca conseguiu determina-lo. Adicionalmente, seu desvio padrão dos valores da função objetivo (DF) foi muito alto e a distancia entre a média (MF) e o valor ótimo foi elevado, indicando que o PSO tem uma precisão pobre e pouca estabilidade para o cálculo de uma boa solução. Quando um enxame de dez mil partículas foi empregado na função B20D, a confiança aumentou para 90% (com aproximadamente 800000 chamadas da função), este fato evidencia a importância de uma apropriada seleção dos parâmetros heurísticos.

O desempenho do SA em duas (funções B20D e VS) das três funções teste foi bom, mas com um alto número de avaliações da função, quando comparado com o PSOS. Em média, o SA empregou entre 13 e 120 vezes mais chamadas da função do que o PSOS. No problema R2D, o SA

apresentou confiança moderada (95%) e em algumas rodadas ele falhou completamente (ver Tabelas 2, 3 e 4). O algoritmo SA apresentou um comportamento estável e teve alta precisão nos exemplos estudados.

O algoritmo genético testado, apresentou menor precisão e estabilidade que o PSOS e o SA nas três funções teste apresentadas. Seu custo computacional, na função VS, foi superior ao do PSOS, em média empregou 10 vezes mais avaliações da função objetivo que o PSOS. Nas funções R2D e B20D e VS o AG apresentou baixa confiança o qual representa um problema para seu emprego direto em problemas de detecção de dano.

Em todos os exemplos considerados, o PSOS nunca fracassou em achar o mínimo global dentro da precisão especificada. Adicionalmente, o desvio padrão (DF) dos resultados foi muito baixo e sua média (MF) ficou muito próxima do valor ótimo das funções analisadas, indicando que as soluções obtidas pelo algoritmo proposto (PSOS), têm alta estabilidade e precisão. Além disto, o número de avaliações da função (ver PC, MC, e DC nas Tabelas 2, 3 e 4) foi sempre menor para o PSOS do que para o SA.

Outro aspecto relevante, evidenciado pelos testes, é que a convergência do algoritmo híbrido proposto não depende de uma boa estimativa inicial ou de uma seleção criteriosa dos parâmetros heurísticos, que nos outros algoritmos devem ser definidos pelo usuário.

Para o PSOS, o desvio padrão e a média dos valores estimados para os parâmetros heurísticos (N , w , C_1 , C_2) são apresentados na Tabela 5. Finalmente, rodadas que atingiram valores de MF, PF ou MF dentro da precisão definida ($1E-4$), foram substituídos pelos valores ótimos analíticos (ver Tabelas 2, 3 e 4).

Tabela 5. Parâmetros heurísticos obtidos pelo PSOS. MV: Média; SD: Desvio Padrão.

| Parâmetro | Venter | | R2D | | B20D | |
|-----------|--------|-------|-------|--------|-------|-------|
| | MV | SD | MV | SD | MV | SD |
| N | 14 | 3,182 | 235 | 85,358 | 274 | 57,49 |
| w | 1,156 | 0,414 | 0,869 | 0,316 | 0,83 | 0,173 |
| C_1 | 1,354 | 0,527 | 1,327 | 0,55 | 2,184 | 0,281 |
| C_2 | 1,384 | 0,578 | 1,209 | 0,518 | 1,539 | 0,38 |

N =número de partículas; w = inércia; C_1 = parâmetro cognitivo; C_2 =parâmetro social.

As médias de w , C_1 , C_2 obtidas mediante o PSOS (ver Tabela 5), estão de acordo com os valores reportados na literatura (ALRASHIDI; EL-HAWARY, 2006; VENTER; SOBIESZCZANSKI-SOBIESKI, 2003). O PSOS selecionou valores de $C_2 \approx C_1$, exceto para a função B20D, indicando que cada partícula põe igual confiança no enxame do que nela mesma.

Finalmente, vale a pena salientar, que a alta estabilidade, precisão e confiança do PSOS é uma característica desejável quando se trata com problemas de detecção de dano, onde a presença do dano não é conhecida *a priori*. Por estas razões, o PSOS e o SA foram empregados nos exemplos de detecção de dano apresentados nos seguintes capítulos.

7 Detecção de Dano Usando os Algoritmos PSOS e SA e Dados Modais: Exemplos Numéricos e Analíticos

7.1 Introdução

Os casos considerados nesta seção são: o problema definido e analisado por Kabe (1985) e por Sako e Kabe (2005) e a detecção de dano numa viga em balanço. A finalidade do primeiro exemplo é obter a matriz de rigidez ajustada a partir dos dados modais do sistema, contaminados por ruído. No segundo exemplo, busca-se determinar a posição e quantificar o dano dentro da viga empregando os algoritmo SA e PSOS. Todos os exemplos foram analisados dez vezes e a melhor rodada, assim como também, as estatísticas das análises foram reportadas. O algoritmo SA foi programado na linguagem FORTRAN (Apêndice A).

7.2 Problema de Kabe: Tolerância ao Ruído

Este é um dos testes mais populares para métodos de ajuste de modelos e detecção de dano. A estrutura é mostrada na Figura 12. Kabe (1985), utiliza dados modais e o método dos multiplicadores de Lagrange

para incorporar as restrições de simetria e de força definidas dentro do procedimento e realizar o ajuste. Este problema é classificado como de difícil solução devido à grande diferença relativa entre as magnitudes dos coeficientes de rigidez e à alta densidade modal do sistema, com diferença máxima de 26,8% entre a primeira e a oitava frequência natural e de 5,2% entre a primeira e a quarta frequência.

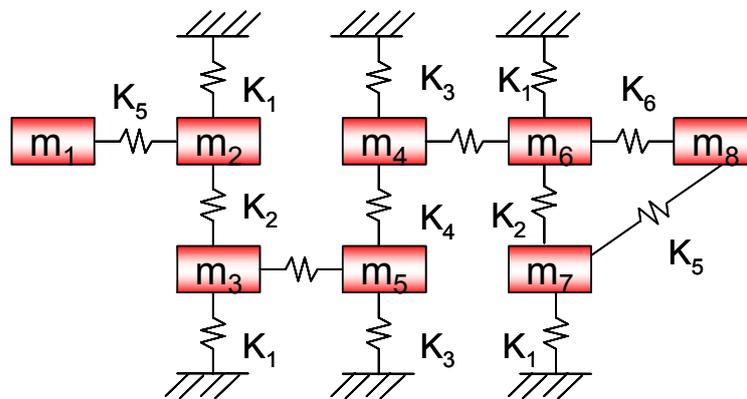


Figura 12. Problema de Kabe.

Neste exemplo, foram escolhidas como variáveis para o problema de otimização os seis coeficientes de rigidez K_i . Os valores exatos dos coeficientes K_i e m_i são dados na Tabela 6. A função objetivo utilizada é a dada pela Equação (40).

Tabela 6. Coeficiente de rigidez e massa.

| K_1 | K_2 | K_3 | K_4 | K_5 | K_6 | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 1000 | 10 | 900 | 100 | 1.5 | 2.0 | | |
| m_1 | m_2 | m_3 | m_4 | m_5 | m_6 | m_7 | m_8 |
| 0.001 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.002 |

Na tabela 7 estão indicados os valores médios do tempo de processamento empregado para solucionar o exemplo (em segundos) (t_m), assim como também: o valor médio do número de avaliações totais da função objetivo (NA_m), o valor médio do número da avaliações da função

objetivo aceitas (NAA_m), o valor médio da função objetivo (f_{objm}), o desvio padrão dos valores da função objetivo (σ), e a confiabilidade (C), que expressa o número de vezes, do total de 10, que o algoritmo convergiu utilizando menos de 40000 avaliações da função objetivo. Também se apresentam, na Tabela 7, os melhores resultados obtidos para o exemplo. O valor t corresponde ao tempo de cálculo (em segundos) da melhor rodada, f_{obj} é o valor da função objetivo, NA é o número total de chamadas da função objetivo e NAA é o número de avaliações da função objetivo aceitas (descidas).

Adicionalmente a Tabela 7 mostra os resultados do ajuste, quando empregados dados experimentais simulados, gerados segundo a metodologia descrita por Kabe (1985) (colunas 1 e 2). As colunas 1 e 2 da Tabela 7 foram calculadas começando a busca desde um ponto inicial diferente (aleatório), e, em todos os 10 testes realizados, o algoritmo convergiu com menos de 20000 avaliações da função objetivo. Comparando os valores dos elementos da matriz de rigidez exata (coluna 5 da Tabela 7) com os valores ajustados utilizando um e dois modos (colunas 1 e 2 da Tabela 6) nota-se que, o SA teve um bom desempenho. Quando comparado com o método de Kabe, pode observar-se que, o SA conseguiu melhores aproximações para 12 dos 16 elementos da matriz $[K]$. Um teste realizado (não apresentado), empregando-se os três primeiros modos analíticos normalizados com relação à massa (sem ruído), reproduziu os valores exatos dos elementos da matriz de rigidez.

O melhor desempenho do SA pode ser explicado pelo fato de a solução de Kabe utilizar, para o cálculo dos multiplicadores de Lagrange (Equação b, Figura 13), a informação modal contaminada com ruído, e, por sua vez

empregar estes multiplicadores para determinar a matriz de rigidez ajustada $[Y]$ (Equação a, Figura 13). As referidas operações matriciais envolvem somas e multiplicações da matriz de formas modais e de freqüências naturais medidas, propagando-se assim os erros de medição para a solução como mostrado na Figura 13. A abordagem direta mediante o SA, e de forma geral, mediante métodos heurísticos, evita os passos intermediários e impede a propagação do erro devido ao uso de funções objetivo do tipo construídas a partir das Equações c, d e e da Figura 13.

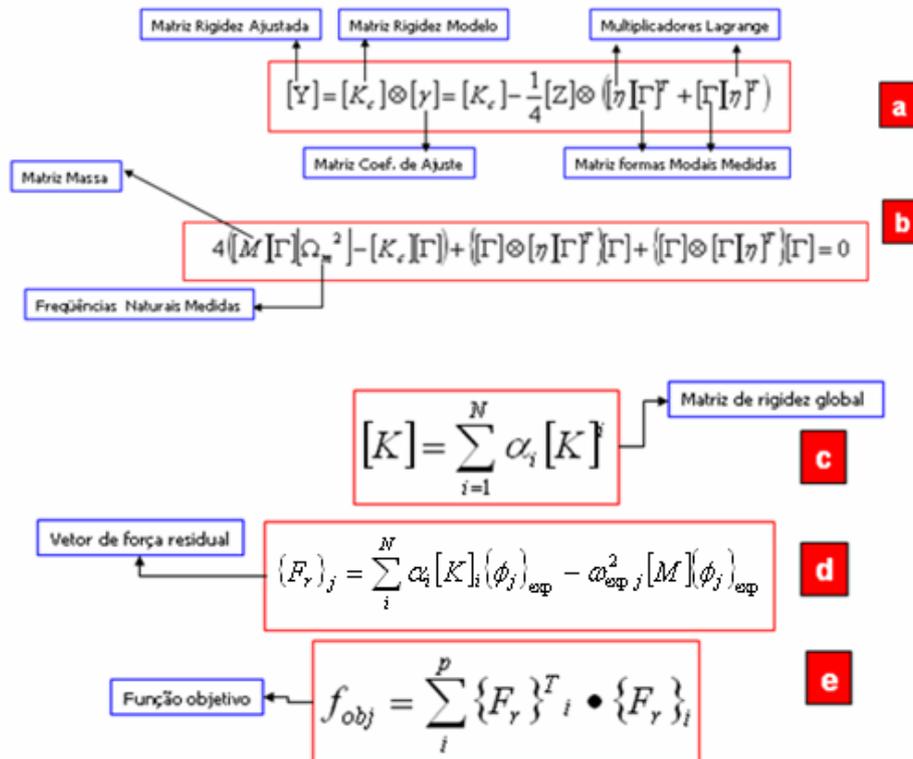


Figura 13. Comparação entre as equações da solução de Kabe e a solução heurística.

Os valores de temperatura inicial (T_0), de redução de temperatura (θ) e de tolerância (ξ), utilizados no algoritmo SA para realizar o exemplo

apresentado (Tabela 7) foram de 50, 0,5 e 1E-6, respectivamente. Um valor muito alto de T_0 pode aumentar o tempo de computação sem adicionar precisão aos cálculos.

Tabela 7. Resultados do ajuste. Problema de Kabe (1985).

| Elem. Matriz [K] | Este trabalho | | Kabe (1985) | | Valor Exato |
|------------------------|---------------|----------|-------------|---------|----------------|
| | [1] | [2] | [3] | [4] | [5] |
| | 1 modo | 2 modos | 1 modo | 2 modos | |
| (1,1) | 2,45 | 1,53 | 1,70 | 1,50 | 1,50 |
| (1,2) | -2,45 | -1,53 | -2,10 | -1,40 | -1,50 |
| (2,2) | 1013,52 | 1012,81 | 1030,40 | 1010,90 | 1011,50 |
| (2,3) | -10,00 | -10,45 | -10,10 | -8,00 | -10,00 |
| (3,3) | 1111,07 | 1111,27 | 1276,60 | 1091,00 | 1110,00 |
| (3,5) | -100,00 | -100,00 | -198,60 | -88,80 | -100,00 |
| (4,4) | 1100,98 | 1101,06 | 1235,20 | 1098,10 | 1110,00 |
| (4,5) | -100,00 | -100,00 | -178,60 | -99,60 | -100,00 |
| (4,6) | -100,00 | -100,00 | -198,50 | -99,60 | -100,00 |
| (5,5) | 1100,98 | 1101,06 | 1239,30 | 1094,00 | 1100,00 |
| (6,6) | 1113,07 | 1114,25 | 1279,80 | 1113,50 | 1112,00 |
| (6,7) | -10,00 | -10,45 | -10,00 | -11,90 | -10,00 |
| (6,8) | -2,00 | -2,98 | -4,10 | -3,10 | -2,00 |
| (7,7) | 1013,52 | 1012,81 | 1002,10 | 1013,60 | 1011,50 |
| (7,8) | -2,45 | -1,53 | -2,00 | -2,40 | -1,50 |
| (8,8) | 4,45 | 4,52 | 5,10 | 4,40 | 3,50 |
| NA | 19201 | 19201 | — | — | — |
| NAA | 9012 | 8942 | — | — | — |
| t | 0,391 | 0,391 | — | — | — |
| f_{obj} | 2,17 | 4,60 | — | — | — |
| C | 100% | 100% | — | — | — |
| t_m | 0,38 | 0,37 | — | — | — |
| f_{objm} | 2,17 | 4,60 | — | — | — |
| σ | 2,55E-08 | 9,89E-09 | — | — | — |
| NA_m | 19021,00 | 18721,00 | — | — | — |
| NAA_m | 8960,50 | 8730,80 | — | — | — |

Para a melhor rodada: t corresponde ao tempo de cálculo (em segundos), f_{obj} é o valor da função objetivo. NA é o número total de chamadas da função objetivo e NAA é o número de avaliações da função objetivo aceitas (descidas). Para as 10 rodadas, t_m é o tempo de processamento empregado para solucionar cada exemplo (em segundos). NA_m é o valor médio do número de avaliações totais da função objetivo. NAA_m é o valor médio do número da avaliações da função objetivo aceitas. f_{objm} é o valor médio da função objetivo. σ o desvio padrão dos valores da função objetivo. C é a confiabilidade que expressa o número de vezes, do total de 10, que o algoritmo convergiu utilizando menos de 40000 avaliações da função objetivo.

7.3 Identificação de Dano Viga em Balanço

O caso considerado é o de uma viga em balanço, discretizada com 12 elementos finitos de barra (ver Figura 14). As propriedades da viga são: área da seção = $4E-4 \text{ m}^2$, densidade = 7800 kg/m^3 , comprimento (L) = $0,8\text{m}$, módulo de elasticidade = 200 Gpa e momento de inércia = $2,38E-8 \text{ m}^4$. Os dados modais foram contaminados com ruído aleatório de $\gamma=1$ para as frequências naturais (ω_i) e de $\lambda=2$ para as formas modais (ϕ_i), segundo a Equação (57), onde $\text{ran}(-1,1)$ é uma função que gera números aleatórios no intervalo $(-1,1)$. Os parâmetros para o SA foram: $T_0=5$, $\theta=0,5$ e $\xi=1E-6$

$$\omega_j = \omega_j + \omega_j \frac{\gamma}{100} \text{ran}(-1,1) \quad e \quad \phi_{ij} = \phi_{ij} + \phi_{ij} \frac{\lambda}{100} \text{ran}(-1,1) \quad (57)$$

Neste exemplo, foram usados os oito primeiros modos para realizar o ajuste da matriz de rigidez. A segunda, terceira e quarta colunas correspondem à viga sem dano e a quinta, sexta e sétima colunas à viga danificada no elemento 4 (20% de perda de rigidez) e no elemento 8 (15% de perda de rigidez). Foi suposto que as rotações não foram medidas experimentalmente e, nesta situação, o modelo analítico foi condensado empregando o método de Kidder (1973), para obter os dados que entram no cálculo da função objetivo (ver Equação (40)). Os resultados obtidos são apresentados na Tabela 8.

Tabela 8. Resultados DD viga em balanço. Algoritmo SA.

| α_i | Sem dano Ruído: 1% nas frequências naturais e 2% nas formas modais | | | Com dano: 20% redução de rigidez elemento 4 e 15 % redução de rigidez no elemento 8. Ruído: 1% nas frequências naturais e 2% nas formas modais | | | |
|---------------|---|----------------------------|-----------|---|--------------------------------------|--------------------------------------|--------------------------------------|
| | Teórico | Detecção SA (α_i) | | Teórico | Detecção SA (α_i) 8modos | Detecção SA (α_i) 5modos | Detecção SA (α_i) 4modos |
| | | 8 modos | 4 modos | | | | |
| α_1 | 1.0 | 0.9988 | 0.9900 | 1.0 | 0.9975 | 0.9883 | 0.9988 |
| α_2 | 1.0 | 0.9975 | 0.9943 | 1.0 | 0.9986 | 0.9949 | 0.9996 |
| α_3 | 1.0 | 0.9993 | 0.9886 | 1.0 | 0.9967 | 0.9943 | 0.9882 |
| α_4 | 1.0 | 0.9987 | 0.9860 | 0.8 | 0.7982 | 0.7857 | 0.7971 |
| α_5 | 1.0 | 0.9932 | 0.9877 | 1.0 | 0.9991 | 0.9997 | 0.9849 |
| α_6 | 1.0 | 0.9982 | 0.9785 | 1.0 | 0.9962 | 0.9802 | 0.9845 |
| α_7 | 1.0 | 0.9899 | 0.9813 | 1.0 | 0.9961 | 0.9932 | 0.9847 |
| α_8 | 1.0 | 0.9982 | 0.9777 | 0.85 | 0.8503 | 0.8471 | 0.8260 |
| α_9 | 1.0 | 0.9997 | 0.9480 | 1.0 | 0.9963 | 0.9827 | 0.9909 |
| α_{10} | 1.0 | 0.9857 | 0.9768 | 1.0 | 0.9988 | 0.9996 | 0.9790 |
| α_{11} | 1.0 | 0.9981 | 0.9443 | 1.0 | 0.9964 | 0.9892 | 0.9862 |
| α_{12} | 1.0 | 0.9917 | 0.9716 | 1.0 | 0.9957 | 0.9989 | 0.9571 |
| NA | - | 12001 | 8401 | - | 9601 | 6001 | 5601 |
| t | - | 6.969 | 2.797 | - | 7.298 | 8.257 | 4.562 |
| f_{obj} | - | 1.68E-8 | 1.2491E13 | - | 2.456E12 | 1.287E11 | 4.672E12 |
| C | - | 100% | 90% | - | 100% | 90% | 90% |
| t_m | - | 6.4315 | 3.504 | - | 8.256 | 6.258 | 4.001 |
| f_{objm} | - | 2.2E-8 | 1.3254E13 | - | 3.012E12 | 2.549E11 | 5.687E12 |
| σ | - | 4.63E-9 | 5.1844E11 | - | 9.365E10 | 2.458E9 | 1.386E10 |
| NA_m | - | 10951,2 | 6001,4 | - | 11506,8 | 7845,3 | 7589,1 |

t_m = tempo médio de processamento; NA_m = valor médio do número de avaliações totais da função objetivo; NAA_m = valor médio do número da avaliações da função objetivo aceitas; f_{objm} = valor médio da função objetivo; σ = desvio padrão dos valores da função objetivo; C = Confiabilidade.

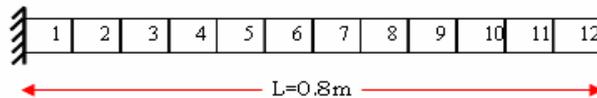


Figura 14. Viga em balanço.

Da Tabela 8 pode concluir-se que, mesmo com adição de ruído, a identificação de dano pôde ser realizada com exatidão aceitável. Para ruídos com $\gamma > 4$ e $\lambda > 10$ não foi possível obter uma avaliação da integridade da viga. O melhor resultado, no caso da viga danificada, foi atingido quando usados

oito modos e uma solução de qualidade aceitável foi alcançada empregando só quatro modos. Neste último cenário, foi calculada uma redução de rigidez de 20,3% para o elemento 4 e de 17,4% para o elemento 8, o que constitui um bom resultado. Por outro lado, a redução de rigidez de 4,3% para o elemento 12, constitui uma indicação duvidosa da presença de dano. Para a viga sem dano, a identificação indica uma redução de rigidez de 5,6% no elemento 11. Este resultado pode dever-se a que uso de quatro modos é insuficiente para calcular uma resposta confiável. Quando empregados três modos (em todos os casos), o procedimento de detecção realizou uma identificação inexata. Dos exemplos apresentados, fica claro que o SA pode ser usado para detectar dano em casos com ruído moderado.

Finalmente, o exemplo anterior foi solucionado mediante o emprego do novo algoritmo proposto nesta pesquisa. Vale salientar que não é necessário definir parâmetros heurísticos para resolver o problema, pois o PSOS é um algoritmo autoconfigurado. Na tabela 9 se apresentam os resultados da detecção. Neste exemplo, pode observar-se que, o PSOS supera o desempenho do SA e que a margem de erro na identificação diminuiu de forma consistente. Por exemplo, no caso da viga danificada, a indicação de dano no elemento 12 diminuiu de 4,3% para 2,7% e os elementos danificados (4 e 8) foram identificados com maior precisão, quando comparados com os resultados da análise feita com o SA.

Tabela 9. Resultados DD Viga em Balanço. Algoritmo PSOS.

| α_i | Sem dano Ruído: 1% nas frequências naturais e 2% nas formas modais | | | Com dano: 20% redução de rigidez elemento 4 e 15 % redução de rigidez no elemento 8. Ruído: 1% nas frequências naturais e 2% nas formas modais | | | |
|---------------|--|------------------------------|-----------|---|-------------------------------|-------------------------------|-------------------------------|
| | Teórico | Detecção PSOS (α_i) | | Teórico | Detecção | Detecção | Detecção |
| | | 8 modos | 4 modos | | PSOS (α_i) 8modos | PSOS (α_i) 5modos | PSOS (α_i) 4modos |
| α_1 | 1.0 | 1.0000 | 1.0000 | 1.0 | 1.0000 | 0.9966 | 0.9987 |
| α_2 | 1.0 | 0.9999 | 0.9943 | 1.0 | 0.9986 | 0.9978 | 0.9986 |
| α_3 | 1.0 | 0.9998 | 1.0000 | 1.0 | 0.9967 | 0.9991 | 0.9891 |
| α_4 | 1.0 | 0.9989 | 0.9860 | 0.8 | 0.8000 | 0.7877 | 0.8081 |
| α_5 | 1.0 | 0.9983 | 0.9877 | 1.0 | 0.9995 | 0.9987 | 0.9943 |
| α_6 | 1.0 | 0.9915 | 0.9785 | 1.0 | 0.9988 | 0.9812 | 0.9885 |
| α_7 | 1.0 | 0.9899 | 0.9813 | 1.0 | 0.9971 | 0.9944 | 0.9918 |
| α_8 | 1.0 | 1.0000 | 0.9777 | 0.85 | 0.8500 | 0.8498 | 0.8599 |
| α_9 | 1.0 | 0.9997 | 0.9899 | 1.0 | 0.9988 | 0.9878 | 0.9918 |
| α_{10} | 1.0 | 1.0000 | 0.9868 | 1.0 | 1.0000 | 0.9981 | 0.9799 |
| α_{11} | 1.0 | 0.9981 | 0.9714 | 1.0 | 0.9964 | 0.9875 | 0.9862 |
| α_{12} | 1.0 | 0.9917 | 0.9834 | 1.0 | 0.9957 | 0.9979 | 0.9761 |
| NA | - | 12001 | 4134 | - | 6640 | 2910 | 2552 |
| t | - | 6.969 | 2.797 | - | 5.317 | 4.761 | 3.277 |
| f_{obj} | - | 0.36E-8 | 0.9741E11 | - | 2.456E11 | 1.007E11 | 3.244E12 |
| C | - | 100% | 100% | - | 100% | 100% | 100% |
| t_m | - | 3.5271 | 1.4659 | - | 7.750 | 3.698 | 3.000 |
| f_{objm} | - | 0.39E-8 | 2.5524E11 | - | 4.238E11 | 2.647E11 | 4.169E12 |
| σ | - | 1.87E-9 | 5.1844E10 | - | 1.961E10 | 0.458E9 | 1.256E10 |
| NA_m | - | 5171,2 | 4305,6 | - | 4705,2 | 4765,9 | 47784,2 |

t_m = tempo médio de processamento; NA_m = valor médio do número de avaliações totais da função objetivo; NAA_m = valor médio do número da avaliações da função objetivo aceitas; f_{objm} = valor médio da função objetivo; σ = desvio padrão dos valores da função objetivo; C = Confiabilidade.

8 Detecção de Dano usando o PSOS e FRFS: Exemplos Numéricos e Analíticos

8.1 Introdução

Os exemplos analisados nesta seção são: uma treliça de dez barras, uma viga fissurada com suas extremidades livres e um sistema analítico não linear (o oscilador de Duffing). Nos dois primeiros exemplos, problemas de detecção de dano, o objetivo foi determinar a posição geométrica do dano e sua extensão a partir das FRFs medidas. No último caso, um problema de identificação, procuram-se os valores dos coeficientes de rigidez e amortecimento utilizando a FRF da primeira ressonância do sistema. Os exemplos foram rodados 10 vezes e os resultados da melhor rodada para cada exemplo, assim como também suas estatísticas, são apresentados.

8.2 Treliça de Dez Barras

Neste exemplo, a resposta da treliça danificada foi simulada mediante redução da rigidez inicial de vários membros da estrutura. A estrutura é mostrada na Figura 15 e suas propriedades são: comprimento (L) = $1m$,

densidade = 7700 kg/m^3 , módulo de elasticidade = 195 GPa , momento de inércia = $3E-8 \text{ m}^4$ e área da seção transversal = $4,2E-4 \text{ m}^2$ (para todos os membros). O dano foi introduzido nos membros dois e oito, mediante redução de sua rigidez para 85% do valor inicial. O amortecimento modal ξ_j foi assumido como sendo igual a 0,01. Ruído aleatório foi adicionado às respostas medidas para simular a influência dos erros de medição. Para a treliça, duas situações foram estudadas: primeiro, as FRFs R_{34} e R_{35} foram usadas para calcular a função objetivo (a excitação esta na direção vertical do nó três, ver Equação (43)) e dois sensores foram usados para medir as FRFs nas direções verticais nos nós quatro e cinco (ver Figura 15); segundo, três FRFs , R_{34} , R_{35} e R_{32} , foram usadas para definir a função objetivo. Em ambos os casos as FRFs contêm as três primeiras freqüências naturais.

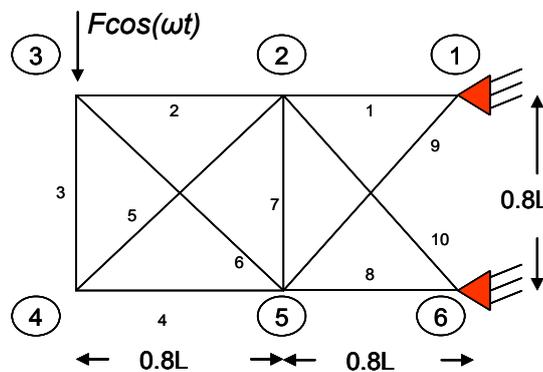


Figura 15. Treliça de dez barras.

A Tabela 10, mostra que uma boa detecção de dano pode ser feita usando só duas FRFs, porém, a redução de rigidez determinada é maior que a introduzida (18,5% calculada contra 15% de redução teórica). Quando o número de FRFs foi três, o procedimento para detecção de dano proposto realizou uma identificação acurada. O tempo médio para a solução do

segundo caso (onde o valor médio do total de avaliações da f_{obj} foi o maior dentre os exemplos apresentados e igual a 36523) foi de aproximadamente 14,6 segundos.

Tabela 10. Resultados do PSOS – Melhor rodada. Treliça.

| Elemento | 15% redução rigidez elementos 2 e 8. Nível de ruído: 3%. R_{34} , R_{35} usadas | | 15% redução rigidez elementos 2 e 8. Nível de ruído: 3%. R_{34} , R_{35} , e R_{32} usadas | |
|------------|---|-----------|--|-----------|
| | Teórico | Deteccção | Teórico | Deteccção |
| 1 | 1.00 | 0.9975 | 1.00 | 1.0000 |
| 2 | 0.85 | 0.8147 | 0.85 | 0.8476 |
| 3 | 1.00 | 0.9749 | 1.00 | 0.9987 |
| 4 | 1.00 | 0.9989 | 1.00 | 0.9862 |
| 5 | 1.00 | 0.9823 | 1.00 | 0.9829 |
| 6 | 1.00 | 0.9990 | 1.00 | 0.9992 |
| 7 | 1.00 | 0.9761 | 1.00 | 1.0000 |
| 8 | 0.85 | 0.8273 | 0.85 | 0.8503 |
| 9 | 1.00 | 0.9881 | 1.00 | 0.9996 |
| 10 | 1.00 | 0.9994 | 1.00 | 1.0000 |
| <i>MEF</i> | 6.27E-5 | 9.98E-3 | 1.71E-3 | 5.31E-3 |
| <i>DF</i> | 5.19E-5 | 3.66E-3 | 5.808E-2 | 6.71E-2 |
| <i>MEC</i> | 10052.41 | 12596.38 | 28233.22 | 36523.95 |
| <i>DC</i> | 4896.62 | 12596.38 | 16900.59 | 14839.09 |
| <i>C</i> | | 100% | | 100% |

média dos valores da função (*MEF*); desvio padrão dos valores da função (*DF*); média do número de chamadas da função (*MEC*); desvio padrão do número de chamadas da função (*DC*); confiabilidade (*C*).

Os valores médios dos parâmetros heurísticos apresentados na Tabela 11 seguem a tendência observada nos exemplos matemáticos (ver Capítulo 6).

Tabela 11. Parâmetros heurísticos obtidos pelo PSOS. Treliça.

| Caso | (1) | (2) |
|-----------|-------|-------|
| Parâmetro | Média | Média |
| N | 84,5 | 79,0 |
| w | 0.267 | 0.189 |
| C_1 | 0.908 | 1.257 |
| C_2 | 1.156 | 1.483 |

N = número de partículas. w = fator de inércia. C_1 e C_2 parâmetros social e cognitivo.

8.3 Viga Livre-Livre com Fissuras

Neste caso, a resposta da viga danificada foi simulada introduzindo uma fissura que permanece aberta durante todo o processo de carga, como explicado na Seção 3.3 (ver Equação (4) e Figuras 2 e 3). A viga foi modelada com doze elementos finitos unidimensionais (Modelo de Euler-Bernoulli). Suas propriedades são: área seção transversal = $1,35E-4 \text{ m}^2$, densidade = 2650 kg/m^3 , comprimento = $1,9 \text{ m}$, módulo de elasticidade = 70 Gpa e momento de inércia = $6,5E-8 \text{ m}^4$. Para este exemplo duas situações foram estudadas: (1) a fissura foi introduzida no elemento 4 com $a_4 = 4E-3$ (altura da fissura) e $x_4 = 0,11$ (posição da fissura dentro do elemento, em coordenadas locais); (2) fissuras simultâneas introduzidas nos elementos 4, 6 e 9, com os seguintes valores: $a_4 = 4E-3$, $a_6 = 3E-3$, $a_9 = 2E-3$ e $x_4 = 0,11$, $x_6 = 0,082$, $x_9 = 0,061$. Um nível máximo de ruído aleatório de 3% foi adicionado às FRFs R_{46} , R_{410} para calcular a função objetivo (Equação (43)). Novamente, em ambos os casos, as FRFs usadas contêm as três primeiras frequências naturais da viga.

Para demonstrar a efetividade do PSOS proposto neste trabalho para identificação de dano, o algoritmo N-M (ver Seção 5.2) foi também utilizado para determinar os parâmetros de dano $\{a, x\}$. Os resultados obtidos com esta técnica são apresentados na Tabela 12 (coluna N-M). O PSOS tem um desempenho melhor do que o N-M, como pode ser observado comparando os valores de C e os valores dos parâmetros $\{a_i, x_i\}$. Nos casos 1 e 2, o método N-M, identificou um número maior de elementos danificados (três e cinco, respectivamente), o qual é falso. Este resultado pobre pode ser explicado

pela incapacidade do algoritmo N-M de fugir de pontos ótimos locais. Por outro lado, tamanhos de fissura menores que $1E-4$, tem um efeito desprezível sobre a resposta da estrutura, portanto, posições x_i e seus valores correspondentes $a_i < 1E-4$ (na Tabela 9), foram igualados a zero, na Tabela 12.

Tabela 12. Resultados do esquema de detecção de dano – Melhor rodada viga livre.

| (a, ξ) | Caso 1 | | | Caso 2 | | |
|----------------------------|---------------------------|---------------------------|--------------------------|---------------------------|---------------------------|--------------------------|
| | <i>PSOS</i> (0% ruído) | <i>PSOS</i> (3% ruído) | <i>N-M</i> (3% ruído) | <i>PSOS</i> (0% ruído) | <i>PSOS</i> (3% ruído) | <i>N-M</i> (3% ruído) |
| a_1 | 0.0 | 0.0 | 1.00E-2 | 0.0 | 0.0 | 6.7E-3 |
| a_2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| a_3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| a_4 | 3.9E-3 | 3.0E-3 | 1.23E-3 | 0.36E-2 | 0.27E-2 | 2.0E-3 |
| a_5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.3E-3 |
| a_6 | 0.0 | 0.0 | 0.0 | 0.29E-2 | 0.24E-2 | 2.99E-3 |
| a_7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| a_8 | 0.0 | 0.0 | 1.90E-4 | 0.0 | 0.0 | 0.0 |
| a_9 | 0.0 | 0.0 | 0.0 | 0.19E-2 | 0.99E-3 | 0.0 |
| a_{10} | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| a_{11} | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| a_{12} | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.8E-3 |
| x_1 | 0.0 | 0.0 | 0.95E-3 | 0.0 | 0.0 | 0.0024 |
| x_2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| x_3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| x_4 | 0.1099 | 0.1130 | 0.1058 | 0.1095 | 0.1132 | 0.1081 |
| x_5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0226 |
| x_6 | 0.0 | 0.0 | 0.0 | 0.0820 | 0.0790 | 0.075 |
| x_7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| x_8 | 0.0 | 0.0 | 0.1385 | 0.0 | 0.0 | 0.0 |
| x_9 | 0.0 | 0.0 | 0.0 | 0.0610 | 0.0587 | 0.0 |
| x_{10} | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| x_{11} | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| x_{12} | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1507 |
| MEF | 1.925E-8 | 7.26E-5 | 1.21E-3 | 2.923E-8 | 4.984E-4 | 9.124E-4 |
| DF | 2.754E-7 | 8.571E-4 | 9.853E-3 | 9.376E-5 | 1.916E-4 | 9.152E-3 |
| MEC | 34001 | 38001 | 22000 | 56768 | 55801 | 24300 |
| DC | 7800.31 | 11238.98 | 15896.67 | 6897.24 | 18756.33 | 12987.28 |
| C | 100% | 100% | 0.00% | 100% | 100% | 0.00% |

média dos valores da função (MEF); desvio padrão dos valores da função (DF); média do número de chamadas da função (MEC); desvio padrão do número de chamadas da função (DC); confiabilidade (C).

Da Tabela 12 pode concluir-se que, a posição da fissura pode ser determinada com maior precisão que seu tamanho. Para o caso 1 com ruído, o erro na posição x_4 foi de 0,09% e o erro em a_4 de 17,5%. Para o caso 2, com ruído, os erros na posição foram $Ex_4= 2,9\%$, $Ex_6= 3,65\%$, $Ex_9=3,77\%$ e os erros no tamanho da fissura foram $Ea_4=30,7\%$, $Ea_6=8\%$, $Ea_9=52,5\%$.

O tempo médio de solução para uma avaliação da função objetivo (solução do problema do auto valor, cálculo das FRFs e avaliação da f_{obj}) para este exemplo foi de 3,03E-4 segundos. Com um total de 55801 avaliações da f_{obj} (maior número de chamadas da função, ver Tabela 12) o tempo total foi de aproximadamente 17 segundos.

Os anteriores resultados confirmam a aplicabilidade do método, já que, uma vez determinada a presença e posição do dano, sua extensão pode ser melhor avaliada mediante o uso de métodos locais de inspeção (por exemplo, inspeção visual, raios X ou métodos de ultra-som).

8.4 Oscilador Não-Linear

A equação de Duffing descreve a oscilação forçada de uma partícula conectada a uma mola não-linear sob a influência de amortecimento viscoso. O movimento do sistema é descrito pela seguinte equação diferencial não-linear de segunda ordem:

$$\ddot{x} + 2\epsilon \dot{x} + \epsilon x^3 = F \cos \Omega t \quad (58)$$

onde, μ é uma constante positiva (relacionada ao amortecimento do sistema), a indica o grau de não-linearidade (relacionado com a rigidez do sistema), ϵ é um fator de escala que indica que o amortecimento e a mola não-linear são uma ordem de magnitude menor que os outros termos e F é a amplitude da

excitação. Neste exemplo, os parâmetros a e μ são iguais a $-1,7$ (*soft spring*) e $0,15$, respectivamente, e a amplitude da excitação F é 0.5 .

A FRF para a primeira ressonância, é determinada pela seguinte equação (NAYFEH; MOOK, 1979):

$$\sigma = \frac{3\alpha a^2}{8\omega_0} \pm \left[-\mu^2 + \frac{F^2}{4\omega_0 a^2} \right]^{1/2} \quad (59)$$

onde, a é a amplitude da resposta, ω_0 é a frequência natural linear ($\omega_0 = \sqrt{k/m} = 1$, neste exemplo) e σ é um parâmetro que descreve a proximidade entre Ω e ω_0 ($\Omega = \omega_0 + \epsilon \sigma$).

O PSOS produz estimativas dos parâmetros α e μ usando a integral da resposta crescente e decrescente do sistema (Figura 16).

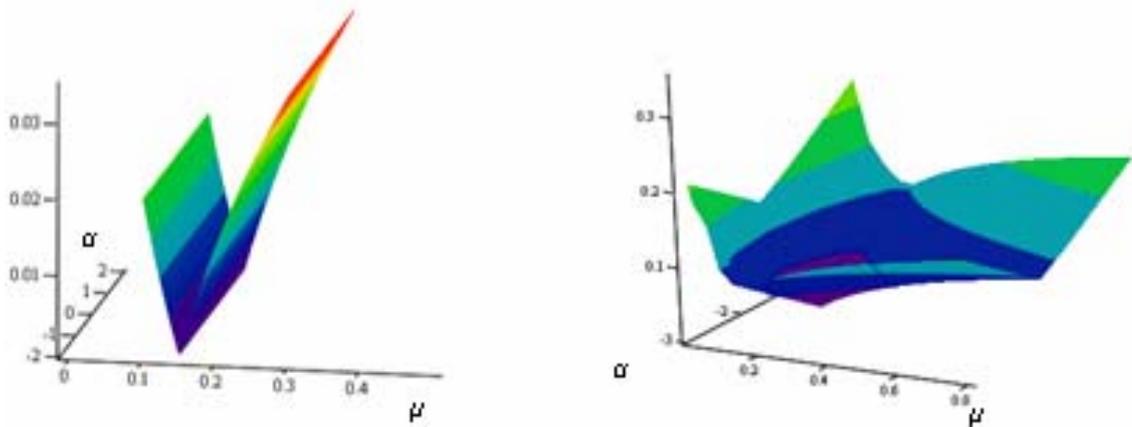


Figura 16. Funções objetivo oscilador de Duffing.

Os resultados do procedimento de identificação são apresentados na Figura 17. Resultados tabelados são apresentados na Tabela 13, a qual revela que a identificação foi acurada e que os valores dos parâmetros heurísticos seguiram a tendência dos exemplos apresentados.

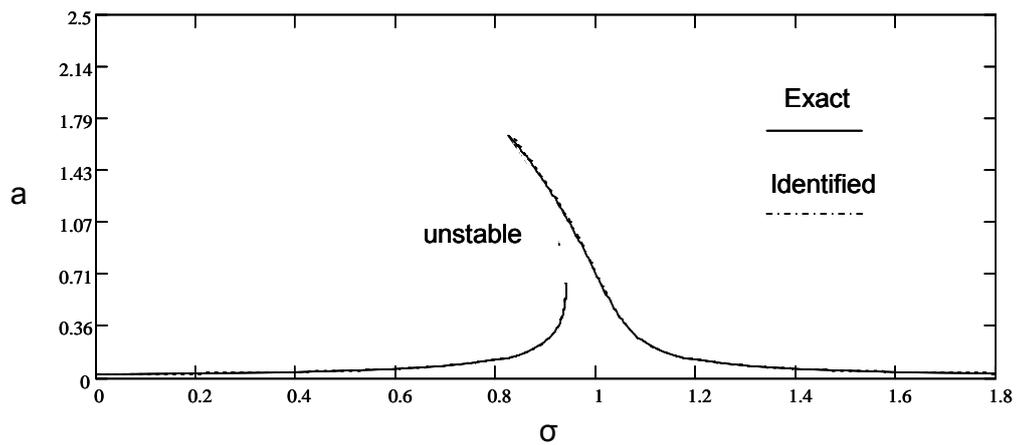


Figura 17. FRFs exata e identificada.

Tabela 13. Resultados da identificação. Oscilador de Duffing.

| a | μ | N | ω | C_1 | C_2 | MEC | MEF |
|--------|-------|-----|----------|-------|-------|--------|---------|
| -1.699 | 0.15 | 38 | 0.17 | 1.29 | 1.58 | 3154.0 | 5.99E-5 |

N = número de partículas. ω = fator de inércia. C_1 e C_2 parâmetros social e cognitivo. MEF = média dos valores da função e MEC = média do número de chamadas da função.

9 Conclusões, Contribuições e Trabalho Futuro

9.1 Conclusões

O algoritmo PSOS foi introduzido neste trabalho. A abordagem se baseia na combinação do PSO básico com o algoritmo Simplex não linear. As principais vantagens deste enfoque são sua alta precisão e estabilidade e sua independência de uma estimativa inicial dos parâmetros heurísticos e de um ponto para iniciar a busca. O algoritmo PSOS foi capaz de determinar o ótimo global de todas as funções teste com alta precisão, estabilidade e confiança, utilizando um número menor de avaliações da função, quando comparado com o *Simulated Annealing*.

Baseado no PSOS, uma metodologia para detecção de dano foi desenvolvida. Este procedimento mostrou um ótimo desempenho em cenários poluídos com ruído e permitiu a identificação correta dos elementos danificados na treliça e nas vigas estudadas. Adicionalmente, o PSOS teve um bom desempenho na identificação das constantes de amortecimento e rigidez do oscilador de Duffing.

A baixa confiança, precisão e estabilidade do algoritmo PSO básico e do método de Nelder-Mead, os fazem impróprios para solucionar problemas de detecção de dano.

A abordagem e solução do problema de Kabe (1985, 2005) realizada neste trabalho, mediante o uso do *Simulated Annealing*, indicam que uma solução direta, com poucas simplificações e aproximações, fornece resultados acurados quando comparada com metodologias clássicas.

O desempenho do AG testado foi inferior ao do PSOS nas funções teste. Uma possível explicação é que o PSOS explora de uma maneira mais completa o espaço de busca e desta forma, sua capacidade de fugir de pontos ótimos locais é reforçada.

A combinação do PSO com outros métodos de otimização, clássicos ou heurísticos, pode melhorar seu desempenho como indicado neste trabalho.

Os tempos de processamento, em todos os exemplos apresentados, foram inferiores a 100 segundos. Espera-se um aumento deste valor em função do aumento do tamanho do problema analisado. Esta dificuldade pode ser superada mediante o emprego de técnicas de processamento em paralelo.

A pesquisa desenvolvida constitui um dos primeiros estudos sobre aplicações e modificações dos algoritmos SA e PSO na área de detecção de dano.

9.2 Contribuições

A seguir são descritas algumas contribuições resultantes dos estudos e simulações numéricas realizados neste trabalho.

O principal aporte deste trabalho é o desenvolvimento de um novo algoritmo híbrido autoconfigurado para detecção de dano.

O algoritmo apresentado é independente dos parâmetros heurísticos e do ponto inicial de busca. Estas características lhe conferem uma vantagem sobre os outros algoritmos existentes, já que, o usuário não precisa realizar inúmeros testes para obter um resultado aceitável, facilitando seu uso na área geral de otimização.

O trabalho apresentado fornece uma demonstração experimental da Estabilidade, Precisão e Robustez do algoritmo proposto (PSOS) mediante análise de múltiplos exemplos, tanto matemáticos como numéricos. Este tipo de análise comparativa raras vezes aparece na literatura e claramente ele fornece elementos para avaliar o desempenho de qualquer algoritmo. O cumprimento desses três critérios (Estabilidade, Precisão e Robustez) é necessário para qualquer procedimento de detecção de dano baseado em modelos, que possa ter aplicações práticas. Este estudo fornece subsídios para futuras formulações de algoritmos híbridos que visem melhorar o desempenho desses métodos, notadamente em relação a sua configuração.

O texto da tese constitui a primeira tentativa de apresentar de forma didática e condensada o material bibliográfico sobre dois temas atuais de grande interesse científico (o problema de detecção de dano e a formulação de métodos heurísticos para otimização) e que hoje em dia se encontram fragmentados na literatura especializada. Também se oferece, nos apêndices, as versões em Fortran dos algoritmos estudados, já que não existem versões comerciais dos mesmos.

9.3 Trabalho futuro

O problema do posicionamento ótimo de sensores e das excitações dentro da estrutura para detecção de dano, empregando o algoritmo PSOS, é um tema de interesse para pesquisas futuras, já que se trata de um problema de otimização multiobjetivo.

Devido ao potencial do algoritmo híbrido desenvolvido, poderão ser implementadas futuras aplicações em outras áreas de otimização.

Dadas suas características, o PSOS pode ser programado em um ambiente paralelo para reduzir o tempo de cálculo em problemas de grande porte (por exemplo, aplicar os algoritmos desenvolvidos para identificação de dano em estruturas reticuladas, isto é, aumentando o tamanho do problema analisado).

Bibliografia

ALRASHIDI, M.R; EL-HAWARY, M.E. (2006). *A survey of particle swarm optimization applications in power system operations*. Electric power components and systems, 34 (12): p.1349-1357.

ALVIN, K. F.; PARK, K. C. (1994). *Second-order structural identification procedure via state-space-based system identification*. AIAA Journal., 32(2), p. 397-406.

ANDERSEN, P; BRINCKER, R; KIERKEGAARD, P. H. (1996). *Theory of covariance equivalent ARMAV models of civil engineering structures*. Proceedings of 14th International Modal Analysis Conference, p. 518-524

AU, F.T.K.; CHENG, Y. S.; THAM, L. G.; BAI, Z. Z. (2003). *Structural damage detection based on a microgenetic algorithm using incomplete and noisy test data*. Journal of Sound and Vibration, 259(5), p. 1081-1094.

BAKHTIARI-NEJAD, F; RAHAI, A; ESFANDIARI, A. (2005). *A structural damage detection method using static noisy data*. Engineering Structures 27, p. 1784-1793.

BANAN, MR; BANAN, MR; HJELMSTAD KD. (1994). *Parameter estimation of structures from static response, I: Computational aspects*. Journal of Structural Engineering, ASCE 120(11), p.3243-59.

BANKS, H.T.; INMAN, D.J.; LEOS, D; WANG, Y. (1996). *An experimentally validated damage detection theory in smart structures*. Journal of Sound and Vibration 191(5), p.859-880

BARUCH, M. (1978-b). *Optimization Procedure to Correct Stiffness and Flexibility Matrices Using Vibration Tests*. AIAA Journal, 16(11), p. 1208-1210.

BARUCH, M. (1997). *Modal data are insufficient for identification of both Mass and Stiffness matrices*. AIAA Journal, 35(11), p.1797-1798.

BATHE, K-J. (1996). *Finite Element Procedures*. Englewood Cliffs, New Jersey, Prentice-Hall.

BEGAMBRE, O.; LAIER, J.E. (2005). *Damage detection using the simulated annealing algorithm*. Proceedings of McMat2005: 2005 Joint ASME/ASCE/SES Conference on Mechanics and Materials. Baton rouge, Louisiana, USA, p.1-4.

BEGAMBRE, O.; LAIER, J.E. (2006). *Procedimento de otimização para ajuste da matriz de rigidez utilizando o algoritmo Simulated Annealing (SA) e dados*

modais. Anais das XXXII Jornadas Sulamericanas de Engenharia Estrutural, p. 2838-2848.

BENNAGE, W.A; DHINGRA, A.K. (1995). *Single and multiobjective structural optimization in discrete-continuous variables using simulated annealing*. International Journal for Numerical Methods in Engineering 38, p. 2753-2773.

BERMAN, A.; NAGY, E. J. (1983) *Improvement of Large Analytical Model Using Test Data*, AIAA Journal, 21, No. 8, pp. 1168-1173.

BLAND, S.M.; KAPANIA, R.K. (2004). *Damage Identification of Plate Structures using Hybrid Genetic-Sensitivity approach*. AIAA Journal, V.43, No2, p.439-442.

BOVSUNOVSKY, A. P.; SURACE, C. (2005). *Considerations regarding superharmonic vibrations of a cracked beam and the variation in damping caused by the presence of the crack*. Journal of Sound and Vibration, 288, p.865-886.

BRUNGER, A.T. (1991). *Simulated Annealing in Crystallography*. Annual Review of Physical Chemistry. 42, p.197-223.

CARLOS, M.F. (2003). *Acoustic emission: Heeding the Warning Sounds from Materials*. ASTM-Standardization News, October 2003. Disponível em:http://www.astm.org/cgi-bin/SoftCart.exe/SNEWS/OCTOBER_2003/carlos_oct03.html?L+mystore+gaci5231+1084370123. Acesso em: 15/04/2004.

CAWLEY, P.; ADAMS, R.D. (1979). *The Locations of Defects in Structures from Measurements of Natural Frequencies*. Journal of Strain Analysis, 14 (2), p.49-57.

CHASE, S. (2001). *The role of smart structures in managing an aging highway infrastructure*. Proceedings of the SPIE conference, New Beach, CA, March 2001. Disponível em:<http://www.tfhr.gov/hnr20/nde/ppt/sld001.htm> Acesso em: 01/01/2004.

CORANA, A.; MARCHESI, M.; MARTINI, C.; RIDELLA, S. (1987). *Minimizing multimodal functions of continuous variables with the Simulated Annealing Algorithm*. ACM Transactions on Mathematical Software. 13, No. 3, p.262-280.

CORNWELL, P.; DOEBLING, S.W.; FARRAR, C. R. (1999). *Application of the Strain Energy Damage Detection method to Plate-like Structures*. Los Alamos National Laboratory, ESA-EA, MS P946. Disponível em:<http://ext.lanl.gov/projects/damage_id/>. Acesso em: 15 jan.2003.

DASCOTTE, E; STROBBE, J.R. (1998). *Updating finite elements models using FRF correlation functions*. Disponível em:<http://www.femtools.com/download/docs/imac99.pdf> Acesso em: 23/09/2005

DOEBLING, S. W.; FARRAR, C. R.; PRIME, M. B; SHEVITZ, D.W. (1996). *Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review*. Los Alamos National Laboratory Report LA-13070-MS. Disponível em:<http://ext.lanl.gov/projects/damage_id/>. Acesso em: 15 jan.2003.

DOEBLING, S.W.; FARRAR, C.R.; PRIME, M.B. (1998). *A Summary Review of Vibration-Based Damage Identification Methods*. The Shock and Vibration Digest, Vol. 30, No. 2, p. 91-105.

DOHERTY, J.E. (1987). *Nondestructive Evaluation*. Chapter 12 in Handbook on Experimental Mechanics. Ed. A.S. Kobayashi. Society For Experimental Mechanics, Inc.

DUFFEY, T. A. et.al.(2000). *Vibration-Based Damage Identification in Structures Exhibiting Axial and Torsional Response*. Los Alamos National Laboratory LA-UR-00-672. Disponível em: <http://ext.lanl.gov/projects/damage_id/>. Acesso em: 15 jan.2003.

EWINS, D. J. (1984). *Modal testing: theory and practice*. Ed. Wiley New York.

FANNING, P.J.; CARDEN, E.P. (2003). *Damage detection based on single-input-single-output measurements*. Journal of Engineering Mechanics. p.202-209.

FANNING, P.J.; CARDEN, E.P. (2004). *Vibration Based Condition Monitoring: a review*. Structural Health Monitoring. Vol 3(4), p.355-377.

FARRAR, C.R., DOEBLING, S.W., DUFFEY, T., A., (2000). *Vibration-Based Damage Detection*. Los Alamos National Laboratory Report MS P-946.

FILHO, L.; ROITMAN, N.;MAGLUTA,C. (2000). *Detecção de Danos Estruturais através de Métodos Diretos de Ajuste de Modelos*. XXIX Jornadas Sudamericanas de Ingenieria, 13-17 Noviembre, Punta Del Este, Uruguay.

FOURIE, P.C; GROENWOLD, A.A. (2002). *The particle swarm optimization algorithm in size and shape optimization*. Struct. Multidisc. Optim. 23, p.259-267.

FOX, C. H. J. (1992). *The Location of Defects in Structures: A Comparison of the use of Natural Frequency and Mode Shape Data*. Proceedings 10th Internationa Modal Analysis Conference, San Diego, California, Vol.1, p.522-528.

FRISWELL, M.I.; MOTTERSHEAD, J. E. (1995). *Finite Element Model Updating in Structural Dynamics*. 1 ed. Dordrecht, The Netherlands, Kluwer Academic Publishers.

FRISWELL, M.I.; PENNY J.E.T, (1997). *Is damaged location using vibration measurements practical?* EUROMECH 365 International Workshop: DAMAS 97, Structural Damage Assessment using Advanced Signal Processing Procedures, Sheffield, UK, June/July 1997.

GARCIA, G.V.; OSEGUEDA, R. (2000). *Combining damage index method and ARMA method to improve damage detection*. Proceedings of the 18th International Modal Analysis Conference. p.668-673.

GOLDBERG, D.E. (1989). *Genetic algorithm in search, optimization and machine learning*. Addison-Wesley Publishing, Reading, U.S.A.

GENOVESE, K.; LAMBERTI, L.; PAPPALETERE, C. (2005). *Equations Improved global-local simulated annealing formulation for solving non-smooth engineering optimization problems*. International Journal of solids and Structures. 42, p.203-237.

GLADWELL, G.M.L. (1997). *Inverse Vibration Problem for Finite-Element Models*. Inverse Problems, 13 p. 311-322.

GOCH, G.; SCHMITZ, B.; KARPUSCHEWSKI, B.; GEERKENS, J.; REIGL, M.; SPRONGL, P.; RITTER, R. (1999). *Review of non-destructive measuring methods for the assessment of surface integrity: a survey of new measuring methods for coatings, layered structures and processed surfaces*. Precision Engineering, 23, p. 9-33.

GOFFE, W.L.; FERRIER, G.D.; ROGERS J. (1994). *Global optimization of statistical functions with simulated annealing*. Journal of Econometrics. 60, p. 65-99.

GUYAN, R.J. (1965). *Reduction of Stiffnes and Mass matrices*. AIAA Journal, 3, (2), p.380.

HAMAMOTO, T., SOMA, S. (2003). *Story Damage Detection of Multistory Buildings Using Genetic Algorithm and Neural Network*. International Workshop on advanced sensor, structural health monitoring and smart structureson. 10-11 November, Keio University, Japan. Disponível em:<http://www.mita.sd.keio.ac.jp/news/workshop/workshop.htm> Acesso em: 24/03/2004.

HAUPT, R.L; HAUPT, SE. (1998). *Practical Genetic Algorithms*. New York, NY: Wiley Interscience.

HEMEZ, F.M; FARHAT, C ; BACHER, E.; VALLAT, S.(1995). *On The efficiency of model updating via genetic algorithm for structural damage detection*. 36th

AIAA/ASME/ASCE/AHS/ASC Structures, structural dynamics and materials conference New Orleans;USA. 10-13 Apr. p. 2792-2801.

HEYLEN, W.; AVITABILE, P., (1998). *Correlation considerations – Part 5*. Proceedings of the 16th International Modal Analysis Conference, February 2-5, Santa Barbara, California, Bethel, CT, Society for Experimental Mechanics, p.207-214.

HJELMSTAD K.D; SHIN, S. (1996). *Crack Identification in a Cantilever Beam from Modal Response*. Journal of Sound and Vibration, 198(5), p.527-545.

HUMAR, J.L. (1990). *Dynamics of Structures*. 1 ed. Englewood Cliffs, New Jersey, Prentice-Hall.

HWANG, S-H; HE, R-S. (2006). *Improving real-parameter genetic algorithm with simulated annealing for engineering problems*. Advances in Engineering Software, 37, p. 406-418.

JAFFE, R.H. (2000). *Random signals for engineers using Matlab® and Mathcad®*. Springer-Verlag New York.

JAMES, G.J; CARNE, T.G; LAUFFER, J.P. (1995). *The natural excitation technique (NExT) for modal parameter extraction from operating structures*. International Journal of Analytical and Experimental Modal Analysis, 10(4), p. 260-277.

JAUREGUI, D.V; FARRAR, C.R. (1996a). *Comparison of Damage Identification Algorithms on Experimental Modal Data from a Bridge*. Proceedings of the 14th International Modal Analysis Conference, p.1423-1429.

JAUREGUI, D.V; FARRAR, C.R. (1996b). *Damage Detection Algorithms Applied to Numerical Modal Data from a Bridge*. Proceedings of the 14th International Modal Analysis Conference, p.119-125.

JONES, K.; TURCOTE, J. (2002). *Finite element model updating using anti resonant frequencies*. Journal of Sound and Vibration, 252(4), p.717-727.

KABE, A.M. (1985). *Stiffness Matrix Adjustment Using Mode Data*. AIAA Journal, 23, (9), p.1431-1436.

KADIRKAMANATHAN, V; SELVARAJAH, K; FELMING, P. (2006). *Stability analysis of the particle dynamics in Particle Swarm Optimizer*. IEEE Transactions on Evolutionary Computation, p.1-11.

KENNEDY J. (1997). *The Particle Swarm: Social adaptation of knowledge*. Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, p. 303-308.

- KENNEDY, J; EBERHART, R. (1995). *Particle swarm optimization*. Proc. IEEE Int. Conf. Neural Networks, 4, p.1942-1948.
- KIDDER, R. L. (1973). *Reduction of Structural Frequency Equations*, AIAA Journal, v.11, n. 6, p.892.
- KIM, H.M.; BARTKOWICZ, T.J. (1993) *Damage Detection and Health Monitoring of Large Space Structures*, Sound and Vibration, 27 (6), p. 12-17.
- KIM, J.-T; RYU, Y-S; CHO, H-M; STUBBS, N. (2003). *Damage Identification in Beam-type Structures: Frequency-based Method vs Mode-shape-based Method*. Engineering Structures, 25 , p.57-67.
- KINCAID, R.K. (1992). *Minimizing distortion and internal forces in truss structures via simulated annealing*. Structural Optimization 4, p. 55-61.
- KIRKPATRICK, S., GELATT, A., VECCHI, M.P., (1983), *Optimization by simulated annealing*. Science, 220, p. 671-680.
- KIRMSHER. P.G, (1944). *The effect of discontinuities on the natural frequencies of beams*. Proceedings of American society of Testing and Materials, 44, p.897-904.
- KOH, C.G; CHEN, Y,F; LIAW, C.-Y. (2003). *A hybrid computational strategy for identification of structural parameters*. Computers and Structures, 81, p. 107-117.
- KRAWCZUK, M.; OSTACHOWICZ, W. M. (1993). *Hexahedral finite element with an open crack*. Finite element in analysis and design. 13, p.225-235.
- KURATA, N., SPENCER, B.F., RUIZ-SANDOVAL, M. (2003). *Risk Monitoring of Buildings Using Wireless Sensor Network*. International Workshop on advanced sensor, structural health monitoring and smart structureson. 10-11 November, Keio University, Japan. Disponível em:<http://www.mita.sd.keio.ac.jp/news/workshop/workshop.htm>. Acesso em: 24/03/2004.
- KWONG, K-S.; LIN R-M. (2005). *Robust finite element model updating using Taguchi method*. Journal of Sound and Vibration, 280(1-2), p.77-99.
- LOCATELLI, M. (2000). *Convergence of a Simulated Annealing Algorithm for continuous global optimization*. Journal of Global Optimization, 18, p. 219-234.
- LAARHOVEN, P.M.J.; AARTS, E.H.L. (1987). *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers.

- LAGARIAS, J.C; REDDS, J.A.; WRIGHT, M.H.; WRIGTH, P.E. (1998). *Convergence properties of the Nealder-Mead Simplex method in low dimensions*, SIAM J. Optim., 9(1), p.112-147.
- LEATH, W. J.; ZIMMERMAN, D. C.; (1993). *Analysis of neural network supervised training with application to structural damage detection*. Proceedings of the 9th Virginia Polytechnic Institute and state university symposium on dynamics and control of large structures, Blacksburg, VA, p.583-594.
- LEE, J.J.; LEE, J.W.; YI,J.H.; YUN, C.B.; JUNG, H.I. (2005). *Neural networks-based damage detection for bridges considering errors in baseline Finite element models*. Journal of Sound and Vibration, 280, p. 555-578.
- LEE, S-Y.; WOOH, S-H. (2005). *Detection of stiffness reductions in laminated composite plates from their dynamic response using the microgenetic algorithm*. Comput Mech. 36, p. 320-330.
- LEONG, W.H.; STASZEWSKI, W.J.; LEE, BC.; SCARPA, F. (2005). *Structural health monitoring using scanning laser vibrometry: III. Lamb waves for fatigue crack detection*. Smart Materials and Structures, 14, p. 1387-1395.
- LIEVEN, N.A.J.; EWINS, D.J. (1988). *Spatial correlation of mode shapes, the coordinate modal assurance criterion (COMAC)*. In Proceedings of the 6th International Modal Analysis Conference, p.690-695.
- LIFSHITZ, J.M.; ROTEM, A. (1969). *Determination of reinforcement unbonding of composites by a vibration technique*. Journal of composite Materials, 3, p.412-423.
- LIU, S-C., AND YAO, J.T.P., (1978). *Structural identification concept*. Journal of the Structural Division, ASCE, 104(ST12), 1845-1858.
- LOU, H.; HANAGUD, S. (1997). *Dynamic learning rate neural network training and structural damage detection*. AIAA Journal, 35(9), p.1522-1527.
- LU, Q.; REN, G.; ZHAO, Y. (2002). *Multiple Damage Location with Flexibility Curvature and relative Frequency Change for Beam Structures*. Journal of Sound and vibration, 253(5), p.1101-1114.
- LUS, H.; De ANGELIS, M.; BETTI, R.; LONGMAN, R. W. (2003). *Constructing second- order models of mechanical systems from identified state space realizations. Part I: theoretical discussion*. Journal of Engineering Mechanics, May, p. 477-488.
- MARES, C.; SURACE, C. (1996). *An application of Genetic Algorithms to identify damage in elastic structures*. Journal of Sound and Vibration, 195(2), p.195-215.

- METROPOLIS, N.; ROSENBLUTH, A.; ROSENBLUTH, M.; TELLER, A.; TELLER, E., (1953). *Equations of state calculations by fast computing machines*, J. of Chem. Physics. 21, p.1087–1092.
- MOITA, J.M; CORREIA, V.,M; MARTINS, P.G.; SOARES, C.M.;SOARES, C.A. (2006). *Optimal design in vibration control of adaptive structures using a simulated annealing algorithm*. Composite Structures, 75, p. 79-87.
- MOSLEM, K.; NAFASPOUR, R. (2002). *Structural Damage Detection by Genetic Algorithms*. AIAA Journal, 40 (7), p.1395–1401.
- NAIR, K.K; KIREMIDJIAN, A. S; LAW, K.H. (2006). *Time series damage detection and localization algorithm with application to the ASCE benchmark structure*. Journal of Sound and Vibration, 291, p.349-368.
- NAYFEHL, A.H.; MOOK, D.T. (1979). *Non-linear oscillations*. New York: John Wiley and Sons.
- NELDER, J.A; MEAD, R. (1965) *A Simplex method for function minimization*, Computer Journal, 7, 308-313, 1965.
- NOCEDAL, J., WRIGHT. S. J. (1999). *Numerical optimization*. New York: Springer.
- O'CALLAHAN, J. (1996). *A procedure for an Improved Reduced System (IRS) Mode*. Proceedings of the 7th International Modal Analysis Conference. New York 1989, p.17-21.
- OSTACHOWICZ , W.M; KRAWCZUK, M. (1990). *Vibration analysis of a cracked beam*. Computer and Structures, 36 (2), p.245-250.
- PANDEY, A.K.; BISWAS, M. (1994). *Damage Detection in Structures Using Changes in Flexibility*. Journal of Sound and Vibration, Vol. 169, 1, p 3-17.
- PANDEY, A.K.; BISWAS, M. (1995). *Experiemental Verification of Flexibility Difference Method for Locating Damage in Structures*. Journal of Sound and Vibration, 184(2), p.311-328.
- PANDEY, A.K.; BISWAS, M.; SAMMAN, M.M. (1991). *Damage Detection from Changes in Curvature Mode Shapes*. Journal of Sound and Vibration, 145(2), p. 321–332.
- PARSOPOULOS, K.E; VRAHATIS, M.N. (2002). *Recent approaches to global optimization problems through particle swarm optimization*, Natural Computing, 1. p.235-306.
- PAWAR, P. M.; GANGULI, R. (2003). *Genetic fuzzy system for damage detection in beams and helicopter rotor blades*. Computer Methods in applied mechanics and engineering, 192, p. 2031-2057.

- PENG, Z.K.; CHU, F.L. (2004). *Application of the wavelet transform in machine condition monitoring and fault diagnosis: a review with bibliography*. Mechanical Systems and Signal Processing, 18, p.199-221.
- PIERCE, S.G.; WORDEN, K.; MANSON, G. (2006). *A Novel information-gap technique to assess reliability of neural networks-based damage detection*. Journal of Sound and Vibration, 293, p. 96-111.
- QI, G.; BARHOST, A.; HASHEMI, J.; KAMALA, G. (1997). *Discrete wavelet decomposition of acoustic emission signals from carbon-fiber-reinforced composites*. Composites Science and Technology, 57(4), p. 389-403.
- RAO, M.A.; SRINIVAS, J.; MURTHY, B.S.N. (2004). *Damage detection in Vibrating bodies using genetic algorithms*. Computers and Structures, 82, p.963-968.
- RAY, M.C.; MALLIK, N. (2004). *Finite element analysis of smart structures containing piezoelectric fiber-reinforced composite actuator*. AIAA Journal, vol. 42, N. 7, July 2004, p.1398-1405.
- REDA TAHA, M.M.; NOURELDIN, A; LUCERO,J.L.; BACA, T.J. (2006). *Wavelet transform for structural health monitoring: a compendium of uses and features*. Structural Health Monitoring, Vol 5(3), p. 267-295.
- REN, W.X; ZHAO, T.; HARIK, I.E. (2004). *Experimental and analytical modal analysis of a steel arch bridge*. Journal of Structural Engineering, ASCE 130(7), p.1-10.
- RUS, G.; CASTRO, E.; GALLEGRO, A.; PEREZ-APARICIO, J.-L.; GARCIA-HERNANADEZ, T. (2004). *Detection and location of damage in rods using wavelets of vibration simulated by the NSM and FEM*. Boller, C. and Staszewki, W.J (eds), Proceedings of the second European Workshop on Structural Health Monitoring, Munich, germany, p. 465-473.
- RYTTER, A., (1993), *Vibration Based Inspection of Civil Engineering*, Ph.D. Dissertation, University of Aalborg, Denmark.
- SAAVEDRA, P.N.; CUITIÑO, L.A. (2001). *Crack detection and vibration behavior of cracked beams*. Computers & Structures, 79, p. 1451-1459.
- SAKO, B.H.; KABE, A.M. (2005). *Direct least-squares formulation of a stiffness adjustment method*. AIAA Journal, 43(2), p. 413-419.
- SALAMA, M; BRUNO, R; CHEN, G-S. ; GARBA, J. (1990). *Optimal placement of excitations and sensors by simulated annealing*. In: Recent advances in multidisciplinary analysis and optimization NASA CP-3031, p. 1441-1458.

SALAWU, O.S. (1997). *Detection of structural damage through changes in frequency: a review*. Engineering Structures, 19(9), p 718-723.

SALAWU, O.S.; WILLIAMS, C. (1994). *Damage Location Using Vibration Mode Shapes*. Proceedings of the 12th International Modal Analysis Conference, p.933-939.

SAMPAIO, R. P. C.; MAIA, N. M. M.; SILVA, J. M. M. (1999). *Damage detection using the frequency-response-function curvature method*. Journal of Sound and Vibration 226(5), p.1029-1042.

SANAYEI, M; SCAMOLPI, S. (1991). *Structures element stiffness identification from static test data*. Journal of Engineering Mechanics Division, ASCE; 117(EM6).

SANG-YOUL, L; SHI-CHANG, W (2005). *Detection of stiffness reductions in laminated composites plates from their dynamic response using the microgenetic algorithm*. Comput Mech, 36, p. 320-330.

SCHUTTE, J.F; GROENWOLD, A.A (2003). *Sizing desing of truss structures using particle swarm particle*. Struct. Multidisc. Optim. 25, p.261-269.

SCHUTTE, J; GROENWOLD, A.A. (2005). *A study of global optimization using particle swarms*. Journal of global Optimization, 31, p. 93-108.

SHI Y, EBERHART RC. (1998). *Parameter selection in particle swarm optimization*. In: Porto VW, Saravanan N, Waagen D, Eiben AE, editors. Evolutionary programming VII, Lecture notes in computer science 1998, p. 591-600, Berlin: Springer.

SINHA, J.K.; FRISWELL, M.I.; EDWARDS, S. (2002). *Simplified models for the locations of cracks in beam structures using measured vibration data*. Journal of Sound and Vibration, 251, 1, p 13-38.

SOHN, H.; FARRAR, C.H. (2001). *Damage diagnosis using time séries analysis of vibration signals* . Smart Materials and Structures. 10, p.446-451.

STASZESWKY, W.J. (2002). *Intelligent signal processing for damage detection in composite materials*. Composites Science and Technology, 62, p. 941-950.

State of-the-art report Monitoring and Safety Evaluation of Existing Concrete Structures (2002). FIB Task Group 5-1. Disponivel em:<http://www.ishmii.org/SHM%20Guidelines/FIB%20Europe%20Task%20Group%205-1%20-%20SHM%20Guidelines/view> Acesso em: 01/100/2005.

STORER, D.M.; TOMLINSON, G.R. (1993). *Recent development in the measurement and interpretation of Higher Order Transfer Functions from non-linear structures*. Mech. Sys. Signal Process. 7, p173-189.

STUBBS, N.; J.-T. KIM; K. TOPOLE. (1992). *An Efficient and Robust Algorithm for Damage Localization in Offshore Platforms*. Proceedings of the ASCE Tenth Structures Congress, p.543-546.

SU, Z; LIN, H-Y; ZHOU, L-M.; LAU, K-T.; YE, L. (2005). *Efficiency of genetic algorithms and artificial neural networks for evaluating delamination in composite structures using fibre Bragg grating sensors*. Smart Materials and structures, 14, p. 1541-1553.

TARANTOLA, A. (2004). *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM .

TAYAL M. (2003). *Particle Swarm Optimization for mechanical design*. Master Thesis, University of Texas at Arlington.

TEUGHEL, A.; MAECK, J.; DE ROECK, G. (2002). *Damage Assessment by FE Model Updating using Damage Functions*. Computers and structures, 80, p.1869-1879.

The 1st International conference on Structural Health Monitoring of Intelligent Infrastructure, SHMII-1'2003.

The 2nd International conference on Structural Health Monitoring of Intelligent Infrastructure, SHMII-2'2005. November 16-18, Shenzhen, China.

The 2nd International Workshop on Structural Health Monitoring of Innovative Civil Engineering Structures. September 22-23, 2004, Winnipeg, Canada.

The 4th World Conference on Structural Control and Monitoring, 4WCSCM 2006. July 11-13. San Diego, California, USA.

The 5th Conference on Damage Assessment of Structures, DAMAS 2003. July 1-3. Southampton, England.

The 6th Conference on Damage Assessment of Structures, DAMAS 2005. July 4-6. Gdansk, Poland.

THOMPSON, W.J, (1943). *Vibration of slender bars with discontinuities in stiffness*. Journal of Applied Mechanics, 17, p.203-207.

TSOU, P.; SHEN, M.H.H, (1994). *Structural damage detection using neural networks*. AIAA Journal, 32, 1, p.176-183.

- VAN OVERSCHEE, P; DE MOOR, B. (1996). *Subspace identification for linear systems: theory, implementation and applications*. Kluwer academic Publishers, Dordrecht, Netherlands.
- VENTER, G.; SOBIESZCZANSKI-SOBIESKI, J. (2003) *Particle Swarm Optimization*. AIAA journal, 41, 8, p.1583-1589.
- WAHAB A. M.M.; DE ROECK, G. (1999). *Damage Detection in Bridges using Modal Curvatures: Application to a Real Damage Scenario*. Journal of Sound and Vibration, 226(2), p.217-235.
- WEAVER, W. JR.; JOHNSTON, P. R. (1987). *Structural Dynamics by Finite Elements*. Ed. Englewood Cliffs, New Jersey, Prentice – Hall.
- WETTER M, WRIGHT J. (2004). *A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization*. Building and Environment; 39, p. 989-999.
- WONG, SV; HAMOUDA, AMS. (2000). *Optimization of fuzzy rules design using genetic algorithm*. Adv Eng Softw, 31, p.251-262.
- WORDEN, K., DULIEU-BARTON, J.M., (2004). *An overview of intelligent fault detection in systems and structures*. Structural health Monitoring, 3(1):85-98.
- WORDEN, K.; SOHN, H.; FARRAR. C.R. (2002). *Novelty detection in a changing environment: regression and interpolation approaches*, Journal of Sound and Vibration, 258(4), 741-761.
- WORDEN, K.; TOMLINSON, G.R. (2001). *Nonlinearity in experimental modal analysis*. Phil.Trans.R. Soc. Lond. A 359, P113-130.
- XXIII International Modal Analysis Conference IMAC 2005. January 31-February 1. Orlando, Fl, USA.
- YUN, C-B.; BAHNG, E.Y., (2000). *Substructural identification using neural networks*. Computers and Structures, 77, 41-52.
- ZANG, C.; IMREGUN, M. (2001). *Structural damage detection using artificial neural networks and measured FRF data reduced via principal component projection*, Journal of Sound and Vibration, Volume 242, Issue 5, 17 May 2001, Pages 813-827.
- ZANG, C.; GRAFE, H.; IMREGUN, M. (2001). *Frequency domain criteria for correlating and updating dynamic finite element models*. Mechanical Systems and Signal Processing 15(1), p.139-155.
- ZENG, P. (1998). *Neural computing in mechanics*. Appl Mech Rev, 51(12), p.173-197.

ZHANG, YF. (2006). *In situ fatigue crack detection using piezoelectric paint sensor*. Journal of Intelligent Material Systems and Structures, 17 (10), p.843-852.

ZIAEI-RAD, S. (2005). *Finite element model updating of rotating structures using different optimization techniques*. Iranian Journal of Science and Technology, Transaction B, Engineering, 29(B6), p.569-585.

ZIMMERMAN, D. C.; SIMMERMACHER, T. (1995). *Model correlation using multiple static load and vibration test*. AIAA Journal, 33, 11, p. 2182-2188.

ZIMMERMAN, D.C.; SIMMERMACHER, T.; KAOUK, M. (2005). *Model correlation and system health monitoring using frequency domain measurements*. Structural Health Monitoring, 4(3), p.213-227.

ZONG, Z; JAISHI, B; GE, J; REN, W.X. (2005). *Dynamic analysis of a half-through concrete-filled tube arch bridge*. Engineering Structures, 27, p 3-15.

ZOU, Y., TONG, L., STEVEN, P. (2000). *Vibration based model dependent damage (delamination) identification and health monitoring for composite structures-a review*. Journal of Sound and Vibration, 230, p. 357-378

Apêndice A: Listagem programa Simulated Annealing

Programado em Fortran segundo o algoritmo proposto por Corana et. al. (1987).

Programa AS

```
USE DFPORT
PARAMETER (N = 2, NFF = 4)
DOUBLE PRECISION XB(N), XU(N), X(N), XOPT(N), C(N), TP(N),
1          FINI(NFF), XP(N), T,YY,s1,s2,Tol, RT, FOPT
INTEGER NP(N), NS, NT, NTA, SD1, SD2, MAF, NAa, NTF, M
REAL(8) elapsed_time, N1,N2,fa
LOGICAL MAX
Character*12 AN
EXTERNAL FCN
elapsed_time=TIMEF()
AN='SAA Rosenbrock 2D'
```

Parâmetros de entrada.

```
N = 2 número de variáveis e NFF = 4 valores das NFF funções critério de
parada (ver seção 4.2.4).
X(N) vetor valores das variáveis
XOPT(N) Valores ótimos da função
FOPT Valor ótimo da função
C(N) vetor para controle do tamanho do passo
TP(N) tamanho do passo
MAX = .FALSE. !MAX = .TRUE. Minimizar ou Maximizar?
Tol = 1.0D-6 Valor Tolerância
RT = 0.85 ! Programa de Esfriamento (redução temperatura)
Iniciar Sementes gerador números aleatórios
CALL RANDOM_SEED()
CALL RANDOM_NUMBER (HARVEST = s1)
N1=s1*100
SD1 = int(N1)
CALL RANDOM_SEED()
CALL RANDOM_NUMBER (HARVEST = s2)
N2=s2*2000
SD2 = int(N2)
NS=15 número de ciclos para ajuste do passo
NT =100 número iterações antes de redução de temperatura
MAF = 1000000 número máximo de avaliações permitidas
T=50000 Temperatura inicial (Sugestões de Corana et al. 1987)
Limites das variáveis
DO J=1,N
    XB(J) =-1001D0 Limite Inferior variáveis
    XU(I) =1001d0 Limite Superior variáveis
    C(I) = 2.0
ENDDO
```

```
10 CONTINUE
    Ponto inicial de busca
    CALL RANDOM_SEED
    Do j=1,N
        CALL RANDOM_NUMBER (HARVEST = s1)
        X(j)= XB(j)+s1*(XU(j)-XB(j))
    End do
DO 20, I = 1, N
```

```

      TP(I) = 1.0
20  CONTINUE

CALL SA(N,X,MAX,RT,Tol,NS,NT,NFF,MAF,XB,XU,C,SD1,
 1      SD2,T,TP,XOPT,FOPT,NAa,NTA,NTF,FINI,XP,NP)

      open(unit=1,file=AN,status='replace')
      write(1,*) 'Resultado SAA : '
      write(1,*) 'Valor Ótimo da Função : '
      write(1,*)(FOPT)
      write(1,*) 'Coordenadas Ponto Ótimo : '
      do I=1,N
      write(1,*)(XOPT(I))
      enddo
      write(1,*) 'Número total de avaliações da Fobj:'
      write(1,*)(NTA)
      write(1,*) 'Número total de avaliações da Fobj aceitas:'
      write(1,*)(NAa)
      elapsed_time=TIMEF()
      WRITE(1,('tempo cálculo:",1F8.4')) elapsed_time
STOP
END

```

```

SUBROUTINE SA(N,X,MAX,RT,Tol,NS,NT,NFF,MAF,XB,XU,C,IPRINT,
 1      SD1,SD2,T,TP,XOPT,FOPT,NAa,NTA,NTF,
 2      FINI,XP,NP)

```

Sobrotina que realiza o procedimento de otimização.

Variáveis externas.

```

      DOUBLE PRECISION X(*), XB(*), XU(*), C(*), TP(*), FINI(*),
 1      XOPT(*), XP(*), T, Tol, RT, FOPT
      INTEGER NP(*), N, NS, NT, NFF, NAa, MAF
 1      NTF, IER, NTA, SD1, SD2
      LOGICAL MAX

```

Variáveis internas.

```

      DOUBLE PRECISION F, FP, P, PP, RATIO
      INTEGER NUP, NED, Ner, NNEW, Lbd, H, I, J, M
      LOGICAL QUIT

```

Funções.

```

      DOUBLE PRECISION EXP
      REAL ALEAT

```

Iniciar o gerador de números aleatórios ALEAT.

```

      CALL IAL(SD1,SD2)

```

Definir valores iniciais

```

      NAa = 0
      NTF = 0
      NTA = 0
      DO 10, I = 1, N
      XOPT(I) = X(I)
      NP(I) = 0

```

```

10  CONTINUE

```

```

      DO 20, I = 1, NFF
      FINI(I) = 1.0D+20

```

```

20  CONTINUE

```

Se temperatura inicial for negativa, avisar.

```

      IF (T .LE. 0.0) THEN
      WRITE(*,*) 'TEMPERATURA INICIAL NEGATIVA'
      RETURN
      END IF

```

```

Se valor inicial das variáveis esta fora dos limites, avisar.
  DO 30, I = 1, N
    IF ((X(I) .GT. XU(I)) .OR. (X(I) .LT. XB(I))) THEN
      WRITE(*,*)'O VALOR INICIAL (X) ESTA FORA DOS LIMITES '
      RETURN
    END IF
30  CONTINUE
Calcular a função no ponto X e voltar com o valor F
  CALL FCN(N,X,F)
Minimizar função FALSE
  IF(.NOT. MAX) F = -F
  NTA = NTA + 1
  FOPT = F
  FINI(1) = F
Inicia laço principal.
100  NUP = 0
     Ner = 0
     NNEW = 0
     NED = 0
     Lbd = 0
     DO 400, M = 1, NT
       DO 300, J = 1, NS
         DO 200, H = 1, N
           Gera XP, valor teste de X. usa TP para definir XP.
             DO 110, I = 1, N
               IF (I .EQ. H) THEN
                 XP(I) = X(I) + (ALEAT()*2.- 1.) * TP(I)
               ELSE
                 XP(I) = X(I)
               END IF
             Se XP fora do limite, escolhe ponto dentro de RA e testa
               IF((XP(I) .LT. XB(I)) .OR. (XP(I) .GT. XU(I))) THEN
                 XP(I) = XB(I) + (XU(I) - XB(I))*ALEAT()
                 Lbd = Lbd + 1
                 NTF = NTF + 1
               END IF
             110  CONTINUE
           Avalia função em XP e devolve FP
             CALL FCN(N,XP,FP)
             IF(.NOT. MAX) FP = -FP
             NTA = NTA + 1
           Muitas avaliações da função==> termina algoritmo.
             IF(NTA .GE. MAF) THEN
               WRITE(*,*)' MUITAS AVALIACOES DA FUNCAO'
               IF (.NOT. MAX) FOPT = -FOPT
               RETURN
             END IF
           Aceita novo ponto se Valor função diminuir
             IF(FP .GE. F) THEN
               DO 120, I = 1, N
                 X(I) = XP(I)
             120  CONTINUE
               F = FP
               NAa = NAa + 1
               NP(H) = NP(H) + 1
               NUP = NUP + 1
           Novo ponto ótimo.
             IF (FP .GT. FOPT) THEN

```

```

                DO 130, I = 1, N
                XOPT(I) = XP(I)
130             CONTINUE
                FOPT = FP
                NNEW = NNEW + 1
            END IF
Critério de Metropolis para aceitar valor maior da função
            ELSE
                P = EXP((FP - F)/T)
                PP = ALEAT()
                IF (PP .LT. P) THEN
                    DO 140, I = 1, N
                    X(I) = XP(I)
140             CONTINUE
                    F = FP
                    NAa = NAa + 1
                    NP(H) = NP(H) + 1
                    NED = NED + 1
                ELSE
                    Ner = Ner + 1

                END IF
            END IF

200     CONTINUE
300     CONTINUE

```

Ajusta Tamanho do Passo (Ver Corana et. Al (1987)).

```

                DO 310, I = 1, N
                RATIO = DFLOAT(NP(I)) /DFLOAT(NS)
                IF (RATIO .GT. .6) THEN
                    TP(I) = TP(I)*(1. + C(I)*(RATIO - .6)/.4)
                ELSE IF (RATIO .LT. .4) THEN
                    TP(I) = TP(I)/(1. + C(I)*((.4 - RATIO)/.4))
                END IF
                IF (TP(I) .GT. (XU(I)-XB(I))) THEN
                    TP(I) = XU(I) - XB(I)
                END IF
310     CONTINUE

                DO 320, I = 1, N
                NP(I) = 0
320     CONTINUE

```

400 CONTINUE

Checar critério de parada.

```

                QUIT = .FALSE.
                FINI(1) = F
                IF ((FOPT - FINI(1)) .LE. Tol) QUIT = .TRUE.
                DO 410, I = 1, NFF
                IF (ABS(F - FINI(I)) .GT. Tol) QUIT = .FALSE.
410     CONTINUE

```

Terminar o SAA

```

                IF (QUIT) THEN
                    DO 420, I = 1, N
                    X(I) = XOPT(I)
420     CONTINUE
                IF (.NOT. MAX) FOPT = -FOPT
                RETURN

```

```

END IF
Continua novo laço se critério de parada não for satisfeito.
T = RT*T
DO 430, I = NFF, 2, -1
  FINI(I) = FINI(I-1)
430 CONTINUE
F = FOPT
DO 440, I = 1, N
  X(I) = XOPT(I)
440 CONTINUE
GO TO 100
END

```

Subroutine IAL(IJ,KL) Inicia o Gerador de Números aleatórios.

```

  Variáveis sementes:
  0 <= IJ <= 31328
  0 <= KL <= 30081
  real U(97), C, CD, CM
  integer I97, J97
  common /raset1/ U, C, CD, CM, I97, J97
  If( IJ .lt. 0 .or. IJ .gt. 31328 .or.
*   KL .lt. 0 .or. KL .gt. 30081 ) then
    print '(A)', ' O PRIMEIRO NUM SEMENTE DEVE TER VALOR
*ENTRE 0 E 31328'
    print '(A)', ' A SEGUNDA SEMENTE DEVE TER VALOR ENTRE 0 E
*30081'
    stop
  Endif
  i = mod(IJ/177, 177) + 2
  j = mod(IJ , 177) + 2
  k = mod(KL/169, 178) + 1
  l = mod(KL, 169)
  do 2 ii = 1, 97
    s = 0.0
    t = 0.5
    do 3 jj = 1, 24
      m = mod(mod(i*j, 179)*k, 179)
      i = j
      j = k
      k = m
      l = mod(53*l+1, 169)
      if (mod(l*m, 64) .ge. 32) then
        s = s + t
      endif
      t = 0.5 * t
3    continue
    U(ii) = s
2  continue
  C = 362436.0 / 16777216.0
  CD = 7654321.0 / 16777216.0
  CM = 16777213.0 / 16777216.0
  I97 = 97
  J97 = 33
  return
End

```

FUNCTION ALEAT() Gerador de Números Aleatórios

```

  real U(97), C, CD, CM

```

```

integer I97, J97
common /raset1/ U, C, CD, CM, I97, J97
uni = U(I97) - U(J97)
if( uni .lt. 0.0 ) uni = uni + 1.0
U(I97) = uni
I97 = I97 - 1
if(I97 .eq. 0) I97 = 97
J97 = J97 - 1
if(J97 .eq. 0) J97 = 97
C = C - CD
if( C .lt. 0.0 ) C = C + CM
uni = uni - C
if( uni .lt. 0.0 ) uni = uni + 1.0
ALEAT = uni
return
END

```

SUBROUTINE FCN(N,X,F) Sub-rotina para cálculo da função objetivo

```

DOUBLE PRECISION X(N)
REAL*8 F,f1,f2,f3,F13
INTEGER N
Função de Rosenbrock 2D:
F=100*(X(2)-X(1)**2)**2+(1-X(1))**2
RETURN
END

```

Apêndice B: Listagem do programa Particle Swarm Optimization

Programado em Fortran segundo o algoritmo proposto por Kennedy e Eberhart (1995) e Schutte e Groenwold (2003)

Parâmetros de entrada:

N número de partículas
M número de variáveis
JC Critério de parada guarda os JC últimos valores de melhor Fobj.
Ite Número Máximo de iterações permitidas
FCN nesta subrotina, se define a função objetivo

Programa PSO

```
USE DFPORT
PARAMETER (N = 20, M = 2)
PARAMETER (JC = 10, Ite = 100)

DOUBLE PRECISION PL(M), PU(M), PPV(M), PP(M,N), VP(M,N), P(M),
                PPx(M,N), J(N), Jx(N), Jcrit(Ite), Jcritt(JC)
DOUBLE PRECISION PPVV(M), VPmax(M), Xxx(M), A(M,N), B(M,N)
INTEGER Pg,k,LL
REAL*8 w, C1,C2,fa,F,s1,s2,s3,s4
REAL*8 soma, media,ff,desvio,tolerância,Crit
CHARACTER*12 AN
EXTERNAL FCN
AN='RANDOM F13'
```

Parâmetros iniciais:

```
Tolerância=1d-7
C1=2d0
C2=2d0
w=1.4d0
```

Limites das variáveis

```
PL(1) =-50d0 Limite inferior
PL(2) =-50d0
```

```
PU(1) =10d0 Limite superior
PU(2) =10d0
```

Iniciar contador do número de iterações

```
k=0
LL=0 contador critério de parada
```

Definir população inicial de partículas distribuídas de forma aleatória

```
CALL RANDOM_SEED
Do jl=1,N
  do i=1,M
    CALL RANDOM_NUMBER (HARVEST = s1)
    PPV(i)= PL(i)+s1*(PU(i)-PL(i))
    CALL RANDOM_NUMBER (HARVEST = s2)
    PPVV(i)= PL(i)+s2*(PU(i)-PL(i))
  end do
```

```

do l=1,M
    PP(l,jl)=PPV(l)
    VP(l,jl)=PPVV(l)
end do
End do

Definir Limite da velocidade
Do i=1,M
    VPmax(i)=(PU(i)-PL(i))/5d0
End do

Checar Limite de velocidade
Do jj=1,N
    Do i=1,M
        IF (VP(i,jj).GT.VPmax(i)) then
            VP(i,jj)=VPmax(i)
        else
            continue
        End if
    End do
End do
Do jj=1,N
    Do i=1,M
        IF (VP(i,jj).LT.-VPmax(i)) then
            VP(i,jj)=-VPmax(i)
        else
            continue
        End if
    End do
End do

Avaliar função objetivo para toda a população
Do l=1,N
    do i=1,M
        P(i)=PP(i,l)
    End do
    Call FCN(M,P,F)
    J(l)=F
End do

Melhor posição da partícula individual, iteração zero:
PPx=PP
Jx=J

Melhor partícula global
Do ii=1,N
    if (Jx(ii).EQ.Minval(Jx)) then
        Pg=ii
    else
        continue
    end if
End do
Do i=1,M
    Xxx(i)=PP(i,Pg)
end do
10 if(k.LT.Ite) then !if(k.LT.5000) then

Ajuste da velocidade
CALL RANDOM_SEED

```

```

Do jj=1,N
  Do i=1,M
    CALL RANDOM_NUMBER (HARVEST = s3)
    A(i,jj)=w*VP(i,jj)+c1*s3*(PPx(i,jj)-PP(i,jj))
    CALL RANDOM_NUMBER (HARVEST = s4)
    B(i,jj)=c2*s4*(Xxx(i)-PP(i,jj))
  VP(i,jj)=A(i,jj)+B(i,jj)
  end do
End do
Checar Limite de velocidade
Do jj=1,N
  Do i=1,M
    IF (VP(i,jj).GT.VPmax(i)) then
      VP(i,jj)=VPmax(i)
    else
      continue
    End if
  End do
End do
Do jj=1,N
  Do i=1,M
    IF (VP(i,jj).LT.-VPmax(i)) then
      VP(i,jj)=-VPmax(i)
    else
      continue
    End if
  End do
End do
Ajuste da posição
Do jj=1,N
  Do i=1,M
    PP(i,jj)=PP(i,jj)+VP(i,jj)
  end do
End do
Checar Limite da posição
Do jj=1,N
  Do i=1,M
    IF (PP(i,jj).GT.PU(i)) then
      PP(i,jj)=PU(i)
    else
      continue
    End if
  End do
End do
Do jj=1,N
  Do i=1,M
    IF (PP(i,jj).LT.PL(i)) then
      PP(i,jj)=PL(i)
    else
      continue
    End if
  End do
End do
Ajuste melhor individuo: avaliação de Fobj
Do l=1,N
  do i=1,M
    P(i)=PP(i,l)
  End do
Call FCN(M,P,F)

```

```

        J(l)=F
    End do
    Atualiza o melhor individuo e o melhor valor Fobj
    Do i=1,N
        IF (J(i).LT.Jx(i)) then
            Jx(i)=J(i)
            Do jj=1,M
                PPx(jj,i)=PP(jj,i)
            End do
        Else
            continue
        End if
    End Do

    Atualiza o melhor resultado global
    Do jj=1,N
        IF (Jx(jj).EQ.Minval(Jx)) then
            Pg=jj
        else
            continue
        end if
    End do
    Do i=1,M
        Xxx(i)=PP(i,Pg)
    end do
    Crit=Minval(J)  Guarda o conhecimento atual das partículas

    Critério de parada: desvio padrão das ultimas N avaliações da Função objetivo
    if ((LL+1).LT.JC) then
        Jcrit(LL+1)=Crit
    else
        continue
    end if
    if (LL.GE.JC.and.LL.LT.Ite) then
        Jcrit(LL)=Crit
        nj=LL-JC
        do i=1,JC
            Jcritt(i)=Jcrit(i+nj) !guardar os JC últimos resultados no
        end do !vetor Jcritt
        soma=0
        do i=1,JC
            soma=soma+Jcritt(i)
        end do
        media=soma/JC
        ff=0
        do i=1,JC
            ff=(ff+(Jcritt(i)-media)**2)/(10-1)
        end do
        desvio=sqrt(ff)
        if (desvio.LT.tolerancia) then
            go to 20
        else
            continue
        end if
        LL=LL+1
    else
        LL=LL+1
        continue
    end if

```

```

                end if

                k=k+1
                w=w*0.98 Redução linear de inércia
                go to 10
            else
20          end if

```

Grava resultados em arquivo, nome arquivo

```

open(unit=1,file=AN,status='replace')
write(1,*)'Valor Jcritt : '
    do I=1,JC
        write(1,*)(Jcritt(I))
    enddo
write(1,*)'Melhor posicao : '
    do I=1,M
        write(1,*)(Xxx(I))
    enddo
write(1,*)'Melhor Funcao obj : '
    write(1,*)(Minval(Jx))
write(1,*)'Número iteracoes k : '
    write(1,*)(k)

```

End

Cálculo função objetivo Venter, 2003

SUBROUTINE FCN(M,P,F)

```

DOUBLE PRECISION P(M)
REAL*8 F,f1,f2,f3,F13
INTEGER M
!funcao de venter 2D
f1=P(1)**2-100*cos(P(1))**2-100*cos(P(1)**2/30)+P(2)**2
f2=-100*cos(P(2))**2-100*cos(P(2)**2/30)
F=f1+f2+1400d0
return
end

```

Apêndice C: Listagem do programa PSOS

O PSOS foi programado na linguagem Fortran segundo o algoritmo apresentado no Capítulo 5 deste trabalho.

Programa PSOS

```
USE IMSL
INTEGER MM
Parameter(MM=4)
INTEGER IB,MAX
real*8 FT, FVA,X(MM), XG(MM),XLB(MM),XUB(MM),S
CHARACTER*12 AN
External FCN
AN='DUMPOL PSO'
```

```
XG(1)=1.4d0 chute inicial fator redução Inércia w
XG(2)=2d0 chute inicial c1
XG(3)=2d0 chute inicial c2
XG(4)=20d0 chute inicial N
MAX=200
S=1
FT=1d-4
```

CALL DUMPOL(FCN,MM,XG,S,FT,MAX,X,FVA) Cálculo no espaço de parâmetros heurísticos

DUMPOL rotina interna do Fortran baseada no Simplex. O usuário pode programá-la segundo o apresentado na seção 5.2)

```
open(unit=1,file=AN,status='replace')
write(1,*)'***** '
write(1,*)'Resultado final do DUMPOL :'
write(1,*)'Minimo Fobj :'
write(1,*)FVA
write(1,*)'Parametros otimos do PSO :'
do i=1,MM
    write(1,*)X(i)
end do
write(1,*)'Número de avaliacoes Realizadas :'
write(1,*)MAX
End
```

SUBROUTINE FCN(MM,X,FFF) Cálculo no espaço de valores da função objetivo.

Implementa o PSO de acordo com o exposto no capítulo 5 desta tese.

```
use DFPORT
DOUBLE PRECISION X(MM)
REAL *8 FFF
PARAMETER ( M = 2) M = número de variáveis espaço de parâmetros
INTEGER MM
PARAMETER (JC = 20, Ite = 80) JC os n últimos valores de melhor Fobj,
Ite = número máximo de iterações permitidas
REAL*8,dimension(:,:),allocatable::PP,VP,PPx,A,B
REAL *8,dimension(:,:),allocatable::J,Jx
DOUBLE PRECISION PL(M), PU(M), PPV(M),P(M), Jcrit(Ite), Jcrit(JC)
DOUBLE PRECISION PPVV(M), VPmax(M),Xxx(M)
DOUBLE PRECISION Gbest(Ite)
INTEGER Pg,k,LL
REAL*8 w, c1,c2,fa,F,s1,s2,s3,s4
REAL *8 soma, media,ff,desvio,tolerancia,Crit,NH
CHARACTER*12 AN
```

```

EXTERNAL FCNN
AN='DUMPOL PSO'
Parâmetros iniciais:
X2=X(1)      Fator redução Inércia
c1=X(2)      const. c1
c2=X(3)      const. c2
N=int(X(4))  número de partículas N
NH=5
ALLOCATE(PP(M,N),VP(M,N))
ALLOCATE(PPx(M,N))
ALLOCATE(A(M,N),B(M,N))
ALLOCATE(Jx(N),J(N))
tolerância=1d-6

!Limites para função exemplo: Venter
PL(1)=-50d0
PL(2)=-50d0

PU(1)=10d0
PU(2)=10d0

Iniciar contador do número máximo de iterações
k=0
LL=0  contador critério de parada

Gera o conjunto aleatório de partículas iniciais
CALL RANDOM_SEED
Do jl=1,N

    do i=1,M
        CALL RANDOM_NUMBER (HARVEST = s1)
        PPV(i)= PL(i)+s1*(PU(i)-PL(i))

        CALL RANDOM_NUMBER (HARVEST = s2)
        PPVV(i)= PL(i)+s2*(PU(i)-PL(i))

    end do
    do l=1,M
        PP(l,jl)=PPV(l)
        VP(l,jl)=PPVV(l)
    end do
End do

Limite da velocidade
do i=1,M
    !VPmax(i)=(PU(i)-PL(i))/5d0
    VPmax(i)=(PU(i)-PL(i))/NH
End do

Checar Limite de velocidade

Do jj=1,N
    Do i=1,M
        IF (VP(i,jj).GT.VPmax(i)) then
            VP(i,jj)=VPmax(i)
        else
            continue
    end do
end do

```

```

        End if
    End do
End do

Do jj=1,N
    Do i=1,M
        IF (VP(i,jj).LT.-VPmax(i)) then
            VP(i,jj)=-VPmax(i)
        else
            continue
        End if
    End do
End do

```

Avaliar função objetivo para cada uma das N partículas de M variáveis

```

Do l=1,N
    do i=1,M
        P(i)=PP(i,l)
    End do
    Call FCNN(M,P,F)
    J(l)=F
end do

```

Melhor individual iteração zero:

```

PPx=PP
Jx=J
Melhor posição global:
Do ii=1,N
    if (Jx(ii).EQ.Minval(Jx)) then
        Pg=ii
    else
        continue
    end if
End do
Do i=1,M
    Xxx(i)=PP(i,Pg)
end do

```

```

10 IF(k.LT.Ite) then
    Ajuste da velocidade
    CALL RANDOM_SEED
    Do jj=1,N
        Do i=1,M
            CALL RANDOM_NUMBER (HARVEST = s3)
            A(i,jj)=w*VP(i,jj)+c1*s3*(PPx(i,jj)-PP(i,jj))
            CALL RANDOM_NUMBER (HARVEST = s4)
            B(i,jj)=c2*s4*(Xxx(i)-PP(i,jj))
            VP(i,jj)=A(i,jj)+B(i,jj)
        end do
    End do

```

Checar Limite de velocidade

```

Do jj=1,N
  Do i=1,M
    IF (VP(i,jj).GT.VPmax(i)) then
      VP(i,jj)=VPmax(i)
    else
      continue
    End if
  End do
End do

```

```

Do jj=1,N
  Do i=1,M
    IF (VP(i,jj).LT.-VPmax(i)) then
      VP(i,jj)=-VPmax(i)
    else
      continue
    End if
  End do
End do

```

Ajuste da posição das partículas

```

Do jj=1,N
  Do i=1,M
    PP(i,jj)=PP(i,jj)+VP(i,jj)
  end do
End do

```

Checar Limite da posição

```

Do jj=1,N
  Do i=1,M
    IF (PP(i,jj).GT.PU(i)) then
      PP(i,jj)=PU(i)
    else
      continue
    End if
  End do
End do

```

```

Do jj=1,N
  Do i=1,M
    IF (PP(i,jj).LT.PL(i)) then
      PP(i,jj)=PL(i)
    else
      continue
    End if
  End do
End do

```

Atualiza melhor individuo: avaliação de Fobj para cada partícula

```

Do l=1,N
  do i=1,M
    P(i)=PP(i,l)
  End do
  Call FCNN(M,P,F)
  J(l)=F !vetor com os N (Num.Partículas) valores de Fobj
End do

```

Atualiza o melhor valor Fobj do enxame

```
Do i=1,N
  IF (J(i).LT.Jx(i)) then
    Jx(i)=J(i)
    Do jj=1,M
      PPx(jj,i)=PP(jj,i)
    End do
  Else
    continue
  End if
End Do
```

Atualiza o melhor resultado global

```
Do jj=1,N
  IF (Jx(jj).EQ.Minval(Jx)) then
    Pg=jj
  else
    continue
  end if
End do
```

```
Do i=1,M
  Xxx(i)=PP(i,Pg)
End do
```

Crit=Minval(J) é o conhecimento atual das partículas
Gbest(k+1)=Minval(Jx)

```
Critério de parada: desvio padrão das ultimas JC avaliações da Fobj=
  if ((LL+1).LT.JC) then
    Jcrit(LL+1)=Crit
  else
    continue
  end if
```

```
  If (LL.GE.JC.and.LL.LE.Ite) then
    Jcrit(LL)=Crit
    nj=LL-JC
    do i=1,JC
      Jcritt(i)=Jcrit(i+nj) !guardar os JC últimos resultados no
    end do

    soma=0

    do i=1,JC
      soma=soma+Jcritt(i)
    end do

    media=soma/JC

    ff=0
    do i=1,JC
      ff=(ff+(Jcritt(i)-media)**2)/(JC-1)
    end do
    desvio=sqrt(ff)
    !write(1,*)'desvio :'
```

```

!write(1,*)(desvio)

if (desvio.LT.tolerancia) then
    !w=w*0.98
!write(1,*)'w : '
!write(1,*)(w)
    go to 20
    else
    continue
end if

LL=LL+1
else
    LL=LL+1
    continue
end if
k=k+1
!w=w*0.98          Inércia linear
w=X2*w
go to 10
Else
20  end if

GRAVA RESULTADOS EM ARQUIVO
open(unit=1,file=AN,status='replace')
write(1,*)'-----'
write(1,*)'Melhor posicao : '

    do I=1,M
    write(1,*)(Xxx(I))
    enddo
write(1,*)'===== '
write(1,*)'Melhor Funcao obj : '
    write(1,*)(Minval(Jx))
write(1,*)'===== '

write(1,*)'Número iteracoes k : '
    write(1,*)(k+1)
write(1,*)'parametros N, FRI,c1, c2, 1.4 de Iner Inic,w(final) : '
    write(1,*)N
    write(1,*)(X2)
    write(1,*)(c1)
    write(1,*)(c2)
    write(1,*)1.4
    write(1,*)(w)

write(1,*)'          '
write(1,*)'          '

FFF=Minval(Jx)
RETURN
End

```

SUBROUTINE FCNN(M,P,FF) Cálculo da função objetivo

```

DOUBLE PRECISION P(M)
REAL*8 FF,f1,f2,f3
INTEGER M
!funcao de vender 2D

```

```
f1=P(1)**2-100*cos(P(1))**2-100*cos(P(1)**2/30)+P(2)**2
f2=-100*cos(P(2))**2-100*cos(P(2)**2/30)
    FF=f1+f2+1400d0
return
End
```