



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA - CT

CENTRO DE CIÊNCIAS EXATAS E DA TERRA - CCET

**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA E ENGENHARIA DE
PETRÓLEO – PPGCEP**



TESE DE DOUTORADO

**UMA NOVA ABORDAGEM DE OTIMIZAÇÃO HÍBRIDA USANDO
OS ALGORITMOS PSO, NELDER-MEAD SIMPLEX E O DE
CLUSTERIZAÇÃO K-MEANS PARA INVERSÃO COMPLETA DA
FORMA DA ONDA 1D**

RUTINALDO AGUIAR NASCIMENTO

Orientador (in memoriam): Prof. Dr. Liacir dos Santos Lucena

Orientador: Prof. Dr. João Medeiros de Araújo

Coorientador: Prof. Dr. Álvaro Barroca Neto

Natal / RN
Maio / 2023

**UMA NOVA ABORDAGEM DE OTIMIZAÇÃO HÍBRIDA USANDO
OS ALGORITMOS PSO, NELDER-MEAD SIMPLEX E O DE
CLUSTERIZAÇÃO K-MEANS PARA INVERSÃO COMPLETA DA
FORMA DA ONDA 1D**

Natal / RN

Maio / 2023

RUTINALDO AGUIAR NASCIMENTO

**UMA NOVA ABORDAGEM DE OTIMIZAÇÃO HÍBRIDA USANDO
OS ALGORITMOS PSO, NELDER-MEAD SIMPLEX E O DE
CLUSTERIZAÇÃO K-MEANS PARA INVERSÃO COMPLETA DA
FORMA DA ONDA 1D**

*Tese de doutorado apresentada ao programa de
Pós Graduação em Ciência e Engenharia de
Petróleo – PPGCEP da Universidade Federal
do Rio Grande do Norte, pré-requisito parcial
para a obtenção do título de Doutor em Ciência
e Engenharia de Petróleo.*

Aprovado em ____ de _____ de 2023.

Prof. Dr. João Medeiros de Araújo
Orientador - UFRN

Prof. Dr. Álvaro Barroca Neto
Coorientador – UFRN

Prof. Dr. Gilberto Corso
Membro Interno ao Programa - UFRN

Prof. Dr. Hugo Alexandre Dantas do Nascimento
Membro Externo à Instituição - UFG

Prof. Dr. Marcos Vinícius Cândido Henriques
Membro Externo à Instituição - UFERSA

NASCIMENTO, Rutinaldo Aguiar – Uma Nova Abordagem de Otimização Híbrida usando os Algoritmos PSO, Nelder-Mead Simplex e o de Clusterização K-means para Inversão Completa da Forma da Onda 1D. Tese de Doutorado, UFRN, Programa de Pós-Graduação em Ciência e Engenharia do Petróleo. Área de Concentração: Pesquisa e Desenvolvimento em Ciência e Engenharia de Petróleo. Linha de Pesquisa: Física Aplicada à Exploração e à Produção de Petróleo e Gás Natural - FAP, Natal-RN, Brasil.

Orientador (in memoriam): Prof. Dr. Liacir dos Santos Lucena

Orientador: Prof. Dr. João Medeiros de Araújo

Coorientador: Prof. Dr. Álvaro Barroca Neto

RESUMO

A Inversão Completa da Forma da Onda (*Full Waveform Inversion* - FWI) é formulada como um problema de otimização não linear, que tradicionalmente utiliza métodos de minimização local baseados em derivadas para encontrar o campo escalar de propriedades físicas da subsuperfície que melhor represente os dados sísmicos de campo. No entanto, esses métodos possuem um alto custo computacional e uma precisão limitada a mínimos locais, além de sofrerem com uma lenta taxa de convergência (*cycle-skipping*). Portanto, neste trabalho, foi desenvolvido um algoritmo de otimização híbrido de duas fases baseado em algoritmos de otimização livre de derivadas (*Derivative Free Optimization* - DFO). Na primeira fase, utiliza-se a minimização global e a técnica de agrupamento, enquanto na segunda fase é adotada a minimização local. Na Fase 1, foram utilizados o algoritmo de otimização por enxame de partículas (*Particle Swarm Optimization* - PSO) e o K-means. Na Fase 2, foi utilizado o Simplex Adaptativo de Nelder-Mead (*Adaptive Nelder-Mead Simplex* - ANMS). O novo algoritmo híbrido foi denominado de PSO-Kmeans-ANMS, no qual o K-means é responsável por dividir o enxame de partículas em dois *clusters* em cada iteração. Essa estratégia visa equilibrar automaticamente os mecanismos de exploração e exploração do espaço de busca de parâmetros, permitindo encontrar soluções mais precisas e, conseqüentemente, melhorar a convergência. O algoritmo híbrido proposto foi validado no conjunto de 12 funções benchmark e, em seguida, aplicado ao problema FWI 1D. Os resultados do PSO-Kmeans-ANMS foram comparados com os obtidos pelos algoritmos PSO clássico, PSO modificado e ANMS. As métricas utilizadas foram o tempo médio de execução e a taxa de sucesso, que aceitou erros de até $\pm 4\%$ da solução ótima. Em todos os experimentos de validação e na aplicação da FWI, o algoritmo PSO-Kmeans-ANMS apresentou desempenho satisfatório, fornecendo resultados precisos e confiáveis, o que comprova sua robustez e eficiência computacional. Além disso, a aplicação desse algoritmo híbrido na FWI proporcionou uma redução significativa no custo computacional, representando assim um resultado importante e promissor para a área da sísmica.

Palavras-chave: Inversão completa da forma da onda, Otimização livre de derivadas, Custo computacional.

ABSTRACT

Full Waveform Inversion (FWI) is formulated as a nonlinear optimization problem, which traditionally utilizes derivative-based local minimization methods to find the scalar field of physical properties of the subsurface that best represents the field seismic data. However, these methods have a high computational cost and a limited accuracy to local minima, in addition to suffering from a slow convergence rate (Cycle Skipping). Therefore, in this work, a two-phase hybrid optimization algorithm based on Derivative-Free Optimization (DFO) algorithms was developed. In the first phase, global minimization and clustering technique are used, while in the second phase, local minimization is adopted. In phase 1, the Particle Swarm Optimization (PSO) algorithm and the Kmeans were used. In phase 2, the Adaptive Nelder-Mead Simplex (ANMS) was used. The new hybrid algorithm was named PSO-Kmeans-ANMS, in which the K-means is responsible for dividing the swarm of particles into two clusters at each iteration. This strategy aims to automatically balance the exploration and exploitation mechanisms of the parameter search space, allowing to find more accurate solutions and, consequently, improving convergence. The proposed hybrid algorithm was validated on the set of 12 benchmark functions and then applied to the 1D FWI problem. The results of the PSO-Kmeans-ANMS were compared with those obtained by the classic PSO, modified PSO, and ANMS algorithms. The metrics used were the average execution time and the success rate, which accepted errors of up to $\pm 4\%$ of the optimal solution. In all validation experiments and in the application of the FWI, the PSO-Kmeans-ANMS algorithm showed satisfactory performance, providing precise and reliable results, which proves its robustness and computational efficiency. In addition, the application of this hybrid algorithm in the FWI provided a significant reduction in the computational cost, thus representing an important and promising result for the seismic area.

Keywords: Full waveform inversion, Derivative free optimization, Computational cost.

“Digno és, Jeová, nosso Deus, de receber a glória, a honra e o poder, porque criaste todas as coisas, e por tua vontade elas vieram à existência e foram criadas.”
(Apocalipse 4:11)

Dedico a Jeová, à minha esposa Sabrina Loisy, aos meus Pais, Ana Maria e Miguel, e aos meus Irmãos, Rosilene e Rutemberg, que torcem hoje e sempre pelo meu sucesso.

AGRADECIMENTOS

À Universidade Federal do Rio Grande do Norte – UFRN e ao Programa de Pós-Graduação em Ciência e Engenharia de Petróleo – PPGCEP.

À Secretaria de Estado da Educação, da Cultura, do Esporte e do Lazer do Rio Grande do Norte (SEEC/RN) por ter concedido à minha licença para o Doutorado.

Ao Professor e Orientador (in memoriam), Dr. Liacir dos Santos Lucena, por ter confiado em meu potencial, me ajudado na realização deste trabalho e pela oportunidade concedida.

Ao Professor e Orientador, Dr. João Medeiros de Araújo, por ter aceitado o desafio de dar continuidade ao trabalho de orientação, por todo apoio e confiança.

Ao Professor e Coorientador, Dr. Álvaro Barroca Neto, que me ajudou na realização deste trabalho orientando no desenvolvimento da pesquisa.

Aos professores participantes da banca examinadora da qualificação e da defesa, os quais contribuíram dando sugestões e críticas para o enriquecimento deste trabalho.

Ao Professor Dr. Hugo Alexandre Dantas do Nascimento pelas contribuições para a melhoria deste trabalho.

Ao Dr. Yuri Shalom de Freitas Bezerra pela amizade, parceria e contribuições.

Ao Centro Interdisciplinar de Estudos Avançados sobre Sistemas Complexos em Petróleo e Gás (CISCPET), pela estrutura oferecida para a realização de todas as pesquisas e desenvolvimento deste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Aos amigos e a todos que diretamente ou indiretamente contribuíram para a concretização deste trabalho.

Muito obrigado!

SUMÁRIO

Índice de Figuras	xii
Índice de Tabelas	xvi
Símbolos e Abreviações	xviii
1 Introdução	2
2 Aspectos Teóricos	8
2.1 Inversão Completa da Forma da Onda (FWI).....	8
2.2 Processo de otimização	10
2.3 Modelagem do problema sísmico 1D	13
2.3.1 Equação da onda.....	14
2.3.1.1 Condições iniciais e de contorno.....	15
2.3.2 Solução numérica da equação da onda	15
2.4 Otimização livre de derivadas.....	19
2.4.1 Otimização por enxame de partículas.....	20
2.4.2 Otimização por enxame de partículas modificado	25
2.4.3 Otimização com Simplex Adaptativo de Nelder-Mead.....	30
2.5 Algoritmo de agrupamento K-means.....	35

2.6	Amostragem aprimorada do espaço de parâmetros	38
2.6.1	Curva de Hilbert	39
2.7	Literatura relacionada	40
3	Metodologia.....	45
3.1	Algoritmo híbrido	45
3.2	Validação do algoritmo proposto com as Funções Benchmark.....	51
3.2.1	Avaliação de desempenho com a função de Rosenbrook.....	53
3.2.2	Avaliação de desempenho com a função de Rastrigin	57
3.2.3	Análise da influência do tamanho do enxame nos resultados da otimização	60
3.3	Aplicação do algoritmo proposto a FWI 1D.....	66
3.3.1	Parametrização do Modelo	67
3.3.2	Dados sísmicos sintéticos e modelagem.....	69
3.3.3	Parâmetros da otimização híbrida, critérios de parada e recursos Computacionais	70
4	Resultados e Discussões.....	73
4.1	Avaliação do algoritmo híbrido de otimização no FWI.....	73
4.2	Avaliação comparativa dos métodos de otimização aplicados na FWI.....	75
4.2.1	Caso 1	75

4.2.2	Caso 2	81
4.2.1	Caso 3	84
5	Conclusões e Perspectivas Futuras	89
5.1	Conclusões	89
5.2	Perspectivas Futuras	90
	Referências Bibliográficas	92
	Apêndice A	104
	Apêndice B	114

ÍNDICE DE FIGURAS

Figura 2.1 – Processo iterativo FWI para o problema direto e inverso até que o desajuste seja mínimo.....	11
Figura 2.2 – (a) Malha uniforme de nx pontos 1D; (b) Estêncil	16
Figura 2.3 – (a) Os três vetores associados à atualização da velocidade da partícula no algoritmo PSO original; (b) Equilíbrio entre os mecanismos de exploração e exploração.	26
Figura 2.4 – Movimentos definidos no Nelder-Mead Simplex original: (a) Reflexão do ponto x_3 , dada pelo vértice x_r ; (b) Expansão, representada pelo vértice x_e ; (c) Contração externa, dada pelo vértice x_c ; (d) Contração interna, definida pelo vértice x_{cc} ; (e) Encolhimento, representado por um triângulo menor.....	34
Figura 2.5 – Esquema ilustrativo do algoritmo de clusterização K-means. (a) Seleção aleatória dos centroides representados pelas cruzeiras coloridas; (b) até (e) Atribuição dos pontos de dados aos <i>clusters</i> mais próximos e atualização dos centroides correspondentes; (f) Resultados da clusterização com os grupos representados pelas cores azul, verde e vermelho."	37
Figura 2.6 – Exemplos das curvas de Hilbert: (a) Curva de Hilbert em 3D e (b) Curva de Hilbert em 2D com as partículas distribuídas aleatoriamente nas regiões.	40
Figura 3.1 – Detalhes do processo de otimização do PSO-Kmeans-ANMS na Fase 1 para o critério de parada baseado na razão entre os tamanhos dos <i>clusters</i> no instante atual.....	54
Figura 3.2 – Detalhes do processo de otimização do PSO-Kmeans-ANMS na Fase 2.....	55
Figura 3.3 – Detalhes do processo de otimização do PSO-Kmeans-ANMS na Fase 1 para o critério de parada baseado na razão entre os desvios padrão dos valores da função objetivo referentes aos exames atual e inicial.....	57

Figura 3.4 – Detalhes do processo de otimização do PSO-Kmeans-ANMS na Fase 1 para o critério de parada baseado na razão entre os tamanhos dos <i>clusters</i> no instante atual.....	58
Figura 3.5 – Detalhes do processo de otimização do PSO-Kmeans-ANMS na Fase 2.....	59
Figura 3.6 – Comparação entre as simulações para todos os algoritmos.	62
Figura 3.7 – Detalhes das simulações para a função Rosenbrock, apresentando os resultados obtidos por todos os algoritmos para os enxames com 8 (superior) e 36 (inferior) partículas, respectivamente.	62
Figura 3.8 – Detalhes das simulações para a função Rastrigin, apresentando os resultados obtidos por todos os algoritmos para os enxames com 8 (superior) e 36 (inferior) partículas, respectivamente.	63
Figura 3.9 – Modelo de velocidade sísmica 1D.	67
Figura 3.10 – (a) Evolução da propagação do campo de onda $u(x,t)$ e (b) Dados observados \mathbf{d}^{obs} ou traço sísmico sintético registrado em $x_r = 0,15 \text{ du}$	70
Figura 4.1 – (a) Momento do salto da Fase 1 para a Fase 2; (b) Detalhes do <i>cluster</i> 1 (círculos azuis), <i>cluster</i> 2 (círculos azuis preenchidos) e da construção do simplex inicial (tetraedro preto).....	74
Figura 4.2 – Caso 1, (100×20) : (a) Diâmetros dos enxames inicial e final; (b) Tempo de execução das Fases 1 e 2.	75
Figura 4.3 – Caso 1, (100×20) : (a), (b) e (c) todos os resultados do algoritmo ANMS para as variáveis V_1 , V_2 e h_{rf} , respectivamente. (d), (e) e (f) histogramas correspondentes ao modelos bem-sucedidos para cada variável do modelo.....	78

Figura 4.4 – Caso 1, (100×20) : (a), (b) e (c) todos os resultados do algoritmo PSO clássico para as variáveis V_1 , V_2 e h_{rf} , respectivamente. (d), (e) e (f) histogramas correspondentes ao modelos bem-sucedidos para cada variável do modelo..... 78

Figura 4.5 – Caso 1, (100×20) : (a), (b) e (c) todos os resultados do algoritmo PSO modificado para as variáveis V_1 , V_2 e h_{rf} , respectivamente. (d), (e) e (f) histogramas correspondentes ao modelos bem-sucedidos para cada variável do modelo..... 79

Figura 4.6 – Caso 1, (100×20) : (a), (b) e (c) todos os resultados do algoritmo PSO-Kmeans-ANMS para as variáveis V_1 , V_2 e h_{rf} , respectivamente. (d), (e) e (f) histogramas correspondentes ao modelos bem-sucedidos para cada variável do modelo..... 79

Figura 4.7 – Caso 1, (100×20) : Histograma do tempo de CPU para cada algoritmo de otimização, aplicados apenas aos modelos bem-sucedidos..... 80

Figura 4.8 – Caso 1, (100×20) : (a) Comparação entre os algoritmos divididos em: Classe 1, taxas de sucesso; Classe 2, tempos médios de CPU e Classe 3, valores médios da função objetivo $[\phi(m) \times 10^2]$. (b) Valores da função objetivo para os modelos bem-sucedidos. 81

Figura 4.9 – Caso 2, (50×40) : (a) Comparação entre os algoritmos divididos em: Classe 1, taxas de sucesso; Classe 2, tempos médios de CPU e Classe 3, valores médios da função objetivo $[\phi(m) \times 10^2]$. (b) Valores da função objetivo para os modelos bem-sucedidos. 84

Figura 4.10 – Caso 3, (100×10) : (a) Comparação entre os algoritmos divididos em: Classe 1, taxas de sucesso; Classe 2, tempos médios de CPU e Classe 3, valores médios da função objetivo $[\phi(m) \times 10^2]$. (b) Valores da função objetivo para os modelos bem-sucedidos. 86

Figura 4.11 – Comparação entre as simulações para todos os algoritmos e Casos: (a) taxa de sucesso *versus* população e (b) tempo médio de execução *versus* população..... 87

Figura 1 – Detalhes das simulações para a função Ackley.....	104
Figura 2 – Detalhes das simulações para a função Rastrigin.	105
Figura 3 – Detalhes das simulações para a função Beale.	106
Figura 4 – Detalhes das simulações para a função Rosenbrock.	106
Figura 5 – Detalhes das simulações para a função Sphere.	107
Figura 6 – Detalhes das simulações para a função Zakharov.....	108
Figura 7 – Detalhes das simulações para a função Michalewicz.	108
Figura 8 – Detalhes das simulações para a função Styblinski-Tang.	109
Figura 9 – Detalhes das simulações para a função F12.....	110
Figura 10 – Detalhes das simulações para a função F22.....	111
Figura 11 – Detalhes das simulações para a função F2.....	111
Figura 12 – Detalhes das simulações para a função Peaks.....	112
Figura 13 – Funções Benchmark para validação do algoritmo híbrido.....	113

ÍNDICE DE TABELAS

Tabela 3.1 – Intervalos de valores usados para cada parâmetro do PSO em todas as aplicações desta pesquisa.	52
Tabela 3.2 – Resultados do algoritmo PSO-Kmeans-ANMS para a função Rosenbrock.....	56
Tabela 3.3 – Resultados do algoritmo PSO-Kmeans-ANMS para a função Rastrigin.	60
Tabela 3.4 – População <i>versus</i> taxa de sucesso para a função Rosenbrock.	61
Tabela 3.5 – População <i>versus</i> taxa de sucesso para a função Rastrigin.....	61
Tabela 3.6 – Resultados das melhores soluções para todos os algoritmos. Abreviações: <i>sr(%)</i> , taxa de sucesso; <i>rt(s)</i> , tempo médio de execução; <i>aes</i> , número médio de avaliações; <i>rop</i> , resultado aproximado para o ponto ótimo.	63
Tabela 3.7 – Valores exatos dos parâmetros do modelo verdadeiro com seus respectivos limites inferior e superior.	69
Tabela 4.1 – Caso 1, (100×20) : Resultados da taxa de sucesso <i>sr(%)</i> e do tempo médio de execução <i>rt(s)</i> de cada algoritmo de otimização.....	76
Tabela 4.2 – Caso 1, (100×20) : Resultados dos parâmetros do modelo e da função objetivo para a melhor solução obtida em cada algoritmo de otimização.	76
Tabela 4.3 – Caso 1, (100×20) : Resultados do valor médio e do respectivo desvio padrão para os parâmetros dos modelos bem-sucedidos.	77
Tabela 4.4 – Caso 1, (100×20) : Resultados do valor médio e do respectivo desvio padrão para o tempo de CPU e a função objetivo.	77

Tabela 4.5 – Caso 2, (50×40) : Resultados da taxa de sucesso $sr(\%)$ e do tempo médio de execução $rt(s)$ de cada algoritmo de otimização.....	82
Tabela 4.6 – Caso 2, (50×40) : Resultados dos parâmetros do modelo e da função objetivo para a melhor solução obtida em cada algoritmo de otimização.	83
Tabela 4.7 – Caso 3, (100×10) : Resultados da taxa de sucesso $sr(\%)$ e do tempo médio de execução $rt(s)$ de cada algoritmo de otimização.....	84
Tabela 4.8 – Caso 3, (100×10) : Resultados dos parâmetros do modelo e da função objetivo para a melhor solução obtida em cada algoritmo de otimização.	85

SÍMBOLOS E ABREVIACÕES

ANMS – *Adaptive Nelder-Mead Simplex*

BFGS – Broyden-Fletcher-Goldfarb-Shanno

CFL – Critério de Courant-Friedrichs-Lewy

d^* – Dados calculados ou modelados

d^{obs} – Dados observados

DFO – *Derivative-Free Optimization*

FWI – *Full Waveform Inversion*

GA – *Genetic Algorithm*

L-BFGS – Limited Memory BFGS

MEF – Método dos Elementos Finitos

MDF – Método das Diferenças Finitas

NM – Nelder-Mead

PSO – Particle Swarm Optimization

\Re^n – Espaço Euclidiano n dimensional

\Re^t – Espaço Euclidiano t dimensional

SA – *Simulated Annealing*

Capítulo I

Introdução

1 Introdução

Devido à presença das estruturas geológicas do subsolo geometricamente complexas e com altos contrastes de velocidade, a indústria petrolífera tem investido na inovação das técnicas de imageamento para alcançar resultados cada vez mais precisos em relação às propriedades físicas das camadas em subsuperfície. Esses resultados podem ajudar na identificação das regiões propícias à acumulação de petróleo. Portanto, tornam-se indispensáveis a pesquisa e o desenvolvimento ou o aprimoramento computacional de técnicas de inversão de dados sísmicos que possam ser implementadas, contribuindo assim para uma melhor caracterização das feições geológicas.

A exploração sísmica terrestre baseia-se na geração artificial de um campo de onda, podendo utilizar como fonte de energia sísmica uma carga explosiva detonada a uma certa profundidade. Esse campo propaga-se de forma radial no interior da Terra, percorrendo as camadas geológicas, onde é refletido e/ou refratado nas interfaces entre as rochas de diferentes constituições petrofísicas. Parte dessas propagações retornam à superfície e são captadas por receptores, que as convertem em sinais elétricos, sendo posteriormente registrados pelo sismógrafo no sismograma (dados observados) (Thomas, 2001). Esses dados, contendo informações importantes sobre a litologia da região analisada, são utilizados por alguma técnica de inversão sísmica.

Dentre as diversas técnicas, existe a Inversão Completa da Forma da Onda (*Full Waveform Inversion* - FWI), a qual é capaz de imagear as estruturas e camadas geológicas da subsuperfície, estimando com alta precisão os parâmetros elásticos do modelo de velocidade que descreve, com a melhor aproximação possível, os dados observados. Para isso ocorrer, a técnica de inversão é implementada através de um processo iterativo composto por duas etapas: resolução do problema direto (modelagem direta) e resolução do problema inverso (inversão da equação da onda), utilizando-se de um método de otimização para minimizar a função objetivo que quantifica o desajuste entre os dados observados na exploração sísmica e os calculados na modelagem.

Na FWI, o problema direto consiste em resolver a equação da onda a partir de um modelo inicial de velocidade do interior da Terra, com o objetivo de modelar os efeitos de propagação das ondas sísmicas e obter os dados calculados na modelagem, ou seja, o sismograma sintético (Martins, 2015; Lima, 2006). Já o problema inverso, por sua vez, consiste em inverter os dados calculados obtidos pelo problema direto, tentando atualizar de forma iterativa os parâmetros do modelo matemático através de um processo de otimização, até que se alcance um ajuste satisfatório entre os dados calculados e observados, aproveitando ao máximo das informações disponíveis nos dados o que inclui a fase, a amplitude e o tempo de trânsito do campo de ondas sísmicas (Tarantola, 2005; Fichtner, 2011).

Geralmente, a técnica FWI utiliza o método do operador adjunto (Plessix, 2006), que se baseia em uma estratégia de otimização com derivadas, como o Gradiente Descendente, o Gradiente Conjugado e o Quasi-Newton do tipo BFGS (Broyden-Fletcher-Goldfarb-Shanno) com memória limitada (Limited Memory BFGS – LBFGS) (Brossier *et al.*, 2009). Entretanto, essas abordagens apresentam alto custo computacional e precisão limitada a mínimos locais, exigindo, portanto, um modelo inicial próximo ao ótimo global para convergir para, pelo menos, uma solução aproximada. Para lidar com esses problemas, muitas estratégias foram propostas e, recentemente, algoritmos híbridos baseados em otimização livre de derivadas foram aplicados à técnica FWI com a finalidade de tornar o processo de minimização da função custo mais eficiente e preciso na busca pelo mínimo global.

Devido à versatilidade e simplicidade de implementação das técnicas de otimização livre de derivadas, Derivative Free Optimization (DFO), elas têm se tornado amplamente utilizadas nas mais diversas áreas do conhecimento e vêm se consolidando como uma das mais poderosas ferramentas para aplicações industriais (Conn, Scheinberg, Vicente, 2009). No entanto, a maioria das técnicas DFO apresenta um alto custo computacional em sua aplicação ao FWI, devido à exaustiva exploração do espaço de parâmetros, ao elevado número de parâmetros em muitas aplicações e à utilização de modelos iniciais aleatórios.

Apesar da modelagem direta (solução numérica) ser a etapa com maior custo computacional, o custo total de utilização das técnicas DFO é excessivamente alto, devido aos fatores citados e à própria natureza dessas técnicas, que exigem a realização de muitas avaliações da função objetivo. No entanto, é possível reduzir drasticamente o custo

computacional por meio de ajustes precisos dos parâmetros de busca, parametrizações eficientes dos modelos com menor número de parâmetros possíveis e utilização de informações a priori, como modelos iniciais semelhantes aos de métodos locais. Com esses ajustes, as técnicas de otimização livre de derivadas podem se tornar mais competitivas do que os métodos locais.

A prática de utilizar otimização global na inversão sísmica não é recente e tem sido aplicada há bastante tempo. Na última década, algoritmos como o Algoritmo Genético (*Genetic Algorithm*, GA) e a Busca Aleatória Controlada (*Controlled Random Search*, CRS) (Price, 1977) foram utilizados com sucesso e precisão em problemas FWI 1D (Tran, Hiltunen, 2012; Aguiar *et al.*, 2019). O algoritmo Simulated Annealing (SA) foi aplicado ao FWI elástico em diversos problemas (Tran, Hiltunen, 2012a; Datta, Sen, 2016), apresentando excelentes resultados que demonstram a precisão da inversão global e um tempo computacional atrativo para aplicações reais. Devido ao custo computacional excessivo e outras dificuldades práticas para solução completa de problemas complexos em 2D e 3D, o FWI baseado em otimização global foi aplicado para substituir técnicas de análise de velocidade, como o *picking* para obtenção da velocidade NMO e a tomografia de tempo de trânsito. Essas técnicas buscam modelos de velocidade com conteúdo de baixa frequência para servirem de modelo inicial para o FWI convencional em problemas complexos 2D (Datta, Sen, 2016; Sajeve *et al.*, 2016; Sajeve *et al.*, 2017) e 3D (Diouane, *et al.*, 2015).

Um avanço importante foi a introdução de algoritmos híbridos nos quais a principal estratégia é combinar as técnicas de otimização global e local, a fim de explorar de forma mais eficiente o espaço de busca e obter soluções melhores e mais robustas do que os algoritmos de busca global ou local sozinhos. Tais algoritmos não exigem um bom modelo inicial para obter bons resultados e possuem um custo computacional relativamente baixo em comparação aos algoritmos de otimização global (Chunduru *et al.*, 1997).

Pan *et al.* (2009) propuseram um algoritmo híbrido que combina os algoritmos *Simulated Annealing Genetic Algorithm* (SAGA) e a busca local simplex (Nelder e Mead, 1965) aplicado à tomografia de tempo de trânsito 2D, com resultados relevantes. Para o FWI elástico 1D, destaca-se a junção do GA com o amostrador de Gibbs, *Gibbs Sampler* (Aleardi e Mazzotti, 2016), que obteve excelentes resultados para problemas sintéticos e reais. A

aplicação ao FWI de um algoritmo híbrido composto por Busca Aleatória Controlada, CRS, e busca local Simplex para problemas de simples e média complexidade 1.5D e 2D (Gomes *et al.*, 2017) demonstrou alta precisão e custo computacional menor do que os métodos com derivada. Mais recentemente, novas estratégias de hibridização, associando o método do operador adjunto e algoritmos DFO, tornaram o FWI baseado em otimização global viável computacionalmente para aplicação a dados reais e com resultados equivalentes ou melhores do que o FWI baseado apenas em derivada para problemas complexos (Datta *et al.*, 2019; Moura *et al.*, 2020).

No entanto, ainda há uma infinidade de algoritmos de otimização híbridos na literatura da área de otimização pura que ainda não foram aplicados ao FWI. Compreender como esses algoritmos se comportam, bem como suas vantagens e desvantagens, quando aplicados ao FWI, é de grande importância tanto para a pesquisa acadêmica quanto para as aplicações industriais. Isso indica um vasto campo de possibilidades para a pesquisa na área. Na literatura relacionada, é mostrado que os algoritmos híbridos compostos pelos algoritmos PSO e ANMS tiveram maior sucesso em problemas diversos. Além disso, a técnica de clusterização Kmeans é uma das mais bem-sucedidas até o momento.

Portanto, neste trabalho, foi desenvolvido o algoritmo híbrido denominado PSO-Kmeans-ANMS, o qual adota uma nova abordagem de otimização em duas fases. A Fase 1 consiste no processo de busca global, utilizando uma versão modificada do algoritmo de Otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO), e no agrupamento do enxame em dois *clusters* a cada iteração por meio do algoritmo k-means. Essa fase termina quando um dos *clusters* se torna dominante em tamanho ou quando o enxame se torna homogêneo, avaliado segundo a métrica do desvio padrão entre os valores da função objetivo das partículas. Já a Fase 2 consiste na aplicação do algoritmo de busca local, *Simplex* Adaptativo de Nelder-Mead (*Adaptive Nelder-Mead Simplex* – ANMS), com o objetivo de melhorar a solução obtida na fase anterior em uma região próxima do mínimo global.

O objetivo específico dessa hibridização de métodos é conseguir obter soluções precisas com menos avaliações da função objetivo e, conseqüentemente, reduzir o custo computacional referente ao tempo de processamento para resolver o problema de otimização

de uma função de desajuste complexa e custosa, assim como no caso da FWI. Os resultados desta pesquisa indicam que o algoritmo híbrido proposto alcançou esse objetivo de maneira mais eficiente do que os algoritmos PSO clássico, PSO modificado e ANMS.

Para validação e compreensão, inicialmente aplicamos o algoritmo híbrido proposto a cada uma das 12 funções de benchmark de forma individual. O desempenho de convergência desse algoritmo foi medido por meio da taxa de sucesso e do tempo médio de execução, e os seus resultados foram comparados com os obtidos por cada um dos algoritmos PSO clássico, PSO modificado e ANMS.

O restante desta tese está organizado da seguinte forma: no Capítulo II, são apresentados aspectos teóricos, com os conceitos que auxiliam na compreensão deste trabalho; o Capítulo III mostra o esquema do algoritmo híbrido proposto e a metodologia de trabalho; o Capítulo IV apresenta os resultados e discussões do estudo referente ao desempenho do algoritmo de otimização híbrido na Inversão Completa da Forma da Onda; o Capítulo V aborda as conclusões que foram obtidas neste trabalho e traz recomendações para trabalhos futuros. Por fim, são apresentadas as referências bibliográficas utilizadas neste trabalho, seguidas por um apêndice com as descrições das funções de benchmark e os resultados da aplicação do novo método nessas funções.

Capítulo II

Aspectos Teóricos

2 Aspectos Teóricos

Neste capítulo, são descritos brevemente a técnica de Inversão Completa da Forma da Onda (*Full Waveform Inversion* - FWI), as equações que governam o problema da inversão e sua solução numérica, bem como os conceitos fundamentais necessários para um bom entendimento dos algoritmos de otimização, como o PSO clássico, o PSO modificado e o ANMS, e do algoritmo de clusterização K-means, aplicados ao problema inverso da FWI.

2.1 Inversão Completa da Forma da Onda (FWI)

A Inversão Completa da Forma da Onda é um exemplo de um problema inverso em geofísica, que envolve a estimativa de propriedades físicas da subsuperfície a partir de dados sísmicos. Para entender a FWI, é importante entender tanto o problema direto (que envolve a propagação da onda sísmica na subsuperfície e a geração dos dados) quanto o problema inverso associado (que busca obter informações sobre a subsuperfície a partir dos dados sísmicos obtidos).

O problema sísmico direto consiste em resolver a equação da onda e simular os efeitos de propagação do campo de onda sísmica na subsuperfície, a partir de um modelo de velocidade conhecido, gerando assim os dados calculados ou modelados, também conhecidos como sismograma sintético. Em outras palavras, definindo-se \mathbf{m} e \mathbf{G} , obtém-se \mathbf{d}^* , aplicando-se o seguinte sistema linear de forma compacta:

$$\mathbf{G}\mathbf{m} = \mathbf{d}^*, \quad (2.1)$$

onde $\mathbf{m} \in \mathbb{R}^n$ e $\mathbf{d}^* \in \mathbb{R}^t$ são, respectivamente, os vetores dos parâmetros do modelo e dos dados calculados, enquanto \mathbf{G} é o operador matemático do sistema físico que descreve a relação entre os parâmetros do modelo e os dados modelados, ou seja, que determina os dados em função dos parâmetros.

Na FWI, o operador direto \mathbf{G} é não linear, sendo composto pela equação da onda sísmica e suas restrições para o referido problema, ou seja, as condições iniciais e de contorno.

O problema inverso utiliza a resposta física \mathbf{d}^* do problema direto para fazer estimativas dos parâmetros do modelo \mathbf{m} . Ambos são atualizados, em cada iteração dos problemas direto e inverso, até que seja possível obter os dados físicos modelados com uma melhor aproximação aos dados observados em campo. Em outras palavras, consiste em determinar os parâmetros do modelo a partir de um modelo inicial e dos dados observados em campo (Almeida, 2013).

Na resolução do problema sísmico inverso, a função objetivo mais comum usada é a função de mínimos quadrados, baseada na norma euclidiana L_2 . Essa visa quantificar o desajuste entre os dados observados e os dados calculados a partir de um modelo de velocidade sísmica. Assim, o melhor ajuste desses dados é obtido quando a Equação (2.2) atinge seu valor mínimo (Fichtner, 2011).

$$\phi(\mathbf{m}) = \frac{1}{2} \|\mathbf{d}^{\text{obs}} - \mathbf{d}^*\|_2^2. \quad (2.2)$$

Uma solução direta para a Equação (2.2) é obtida quando o gradiente dessa função é igual a zero, que pode ser escrito da seguinte forma:

$$\mathbf{G}^T \mathbf{G} \mathbf{m} = \mathbf{G}^T \mathbf{d}^{\text{obs}}, \quad (2.3)$$

onde \mathbf{G}^T é o conjugado transposto de \mathbf{G} . Reorganizando a Equação acima, obtém-se:

$$\mathbf{m} = \mathbf{H}^{-1} \mathbf{G}^T \mathbf{d}^{\text{obs}}, \quad (2.4)$$

onde se assume que a Hessiana $\mathbf{H} = \mathbf{G}^T \mathbf{G}$ é invertível, mas essa suposição é problemática. Assim como na inversão direta de \mathbf{G} , porque \mathbf{H} é muito grande para a maioria dos problemas práticos, principalmente em termos de memória de computador, além de ser singular, o que significa que não possui uma inversa regular.

Para o problema FWI, a resolução do sistema presente na Equação (2.4) não é possível na maioria dos casos, pois a solução analítica da equação da onda sísmica torna-se muito complexa e geralmente fica limitada aos problemas simples (Bleistein *et al.*, 2001).

2.2 Processo de otimização

As técnicas de otimização, que podem ser globais ou locais, estão entre os diversos métodos de solução de problemas inversos. No caso da FWI, a solução do problema é ajustada originalmente por meio de um processo de otimização de busca local, cuja finalidade é minimizar a função objetivo. Dessa forma, o problema FWI é naturalmente formulado como um problema de otimização não linear da forma:

$$\begin{aligned} &\text{Minimizar } \phi(\mathbf{m}), \\ &\text{Sujeito a } \mathbf{Gm} = \mathbf{d}^*. \end{aligned} \quad (2.5)$$

Na prática, a FWI pode ser vista como um problema de otimização que começa com uma solução inicial arbitrária ou a priori sobre o modelo de subsuperfície, que é aprimorada iterativamente para obter uma solução mais compatível com os dados observados. Isso é feito repetindo uma sequência de etapas, como realizar a modelagem direta, avaliar uma função objetivo, calcular uma direção de melhoria e atualizar o modelo atual. Tradicionalmente, a FWI é baseada no método do operador adjunto que emprega derivadas. Nessa abordagem, para cada iteração, são calculados uma direção de busca e um comprimento de passo para atualizar o modelo (Tarantola, 2005). A fórmula geral de atualização do modelo pode ser definida como:

$$\mathbf{m}^{k+1} = \mathbf{m}^k - \alpha_h \mathbf{h}^k, \quad (2.6)$$

onde k é o número de iterações, α_h é um escalar positivo que determina o comprimento do passo em relação a \mathbf{h} que, por sua vez, representa a direção de busca. Para o método do gradiente descendente, \mathbf{h} é igual ao gradiente da função objetivo, ou seja, $\mathbf{h} = \nabla_{\mathbf{m}} \phi(\mathbf{m}^k) = \frac{\partial}{\partial \mathbf{m}} \phi(\mathbf{m}^k)$. Assim, considerando o ponto de partida como sendo $k=1$, a primeira iteração é da forma:

$$\mathbf{m}^2 = \mathbf{m}^1 - \alpha_h \frac{\partial}{\partial \mathbf{m}} \phi(\mathbf{m}^1). \quad (2.7)$$

Para o método de Gauss-Newton, $\mathbf{h} = \mathbf{H}^{-1} \frac{\partial}{\partial \mathbf{m}} \phi(\mathbf{m})$, onde \mathbf{H} é a matriz Hessiana da função de desajuste (Zhang *et al.*, 2015). Essa abordagem é computacionalmente muito mais “cara” (demorada) para ser calculada.

A Figura 2.1 mostra esquematicamente como a FWI funciona na atualização iterativa de um modelo inicial \mathbf{m}^0 para um modelo estimado \mathbf{m}^* , até que o desajuste entre os dados observados \mathbf{d}^{obs} e calculados \mathbf{d}^* seja mínimo. Ou seja, até encontrar o mínimo da função objetivo $\phi(\mathbf{m})$, dada pela Equação (2.2). A ideia consiste em caminhar iterativamente pelo espaço dos modelos \mathcal{R}^n , de modo que quanto mais \mathbf{d}^* se aproxima de \mathbf{d}^{obs} , no espaço dos dados \mathcal{R}^t , mais \mathbf{m}^* se aproxima do modelo de referência \mathbf{m}^{ref} ou modelo verdadeiro para o problema FWI. Na parte inferior dessa figura, apresenta-se um exemplo da FWI 1D com o modelo de velocidade composto por duas camadas (v_1 e v_2), separadas por um refletor, além dos traços dos sismogramas referentes aos dados observados \mathbf{d}^{obs} e aos calculados \mathbf{d}^* . Espera-se que quando $\mathbf{d}^* \approx \mathbf{d}^{\text{obs}} \Rightarrow \mathbf{m}^* \approx \mathbf{m}^{\text{ref}}$.

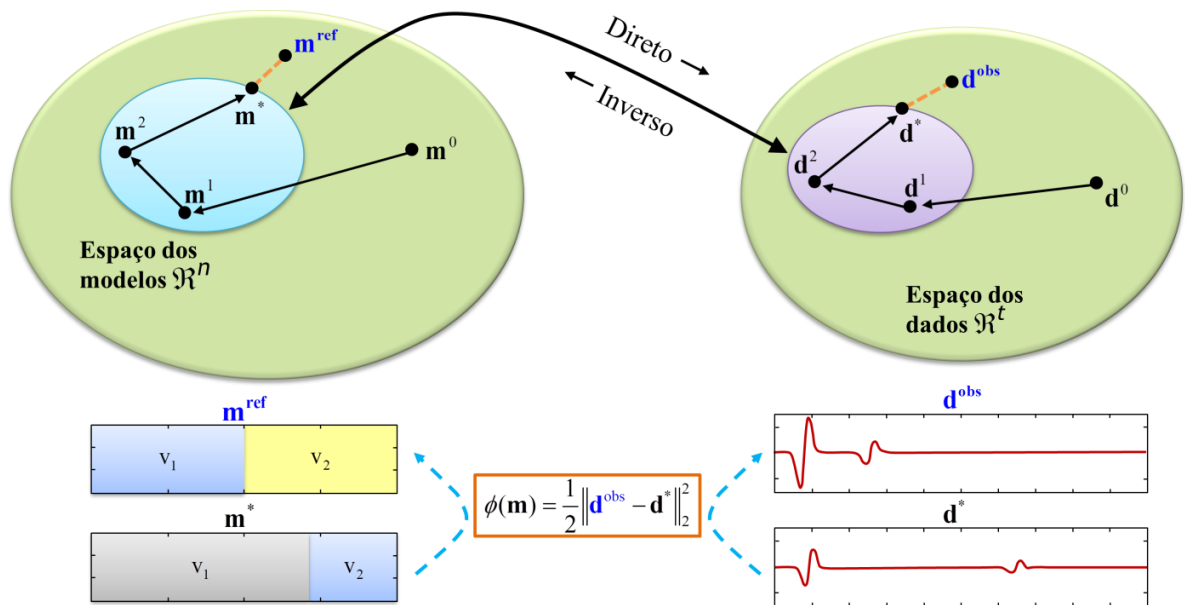


Figura 2.1 – Processo iterativo FWI para o problema direto e inverso até que o desajuste seja mínimo.

O alto custo computacional associado à FWI tradicional ocorre principalmente devido ao método do operador adjunto, que faz uso da propagação reversa do campo de onda para

calcular o gradiente da função objetivo de uma forma eficiente (Plessix, 2006). Isso pode exigir um grande número de iterações para encontrar uma solução confiável. No entanto, esse processo utiliza métodos de busca local que podem ficar presos em mínimos locais, afetando a qualidade da solução. Portanto, os métodos de otimização global podem ser utilizados para superar essa limitação, buscando soluções de forma mais eficiente e com melhor taxa de convergência.

Além disso, uma das principais aplicações desse método de otimização em problemas de FWI é encontrar modelos iniciais confiáveis e, em seguida, usar algum método baseado em gradiente para obter uma estimativa do modelo final. Os algoritmos de otimização global apresentam muitas vantagens neste tipo de aplicação, pois não necessitam de modelos iniciais com alguma característica da solução do problema e podem escapar de mínimos locais que geralmente impedem os métodos que são baseados em gradientes de prosseguir na busca pela solução ideal.

Mais precisamente, a etapa da otimização do problema FWI pode ser tratada como um problema de programação não linear. Na sua forma canônica, é escrito como:

$$\begin{aligned} &\text{Minimizar } \phi(\mathbf{m}) \\ &\text{Sujeito a } \begin{cases} \mathbf{G}\mathbf{m} = \mathbf{d}^* \\ L_{inf_j} \leq m_j \leq L_{sup_j}, \quad j = 1, 2, \dots, n. \end{cases} \end{aligned} \quad (2.8)$$

Nesse contexto, o número de parâmetros do modelo considerado é igual a n , onde L_{inf_j} e L_{sup_j} são as restrições associadas aos limites inferior e superior para cada parâmetro m_j do modelo \mathbf{m} .

Na FWI usando otimização global, a dimensão do espaço de busca, ou seja, o número de parâmetros do modelo que serão otimizados, influencia drasticamente a precisão e a capacidade dos algoritmos em encontrar uma solução ótima. Essa dimensão pode até inviabilizar a busca no espaço de soluções (Aleari e Mazzotti, 2016; Curtis e Lomax, 2001). Por essa razão, a maioria das abordagens aplicadas ao problema FWI faz uso de algum tipo de reparametrização do modelo geológico, a fim de reduzir o número de parâmetros (incógnitas)

e, conseqüentemente, diminuir o custo computacional e permitir a aplicação dos mais diversos algoritmos.

A principal estratégia para otimizar o uso de recursos computacionais utilizados em problemas 2D é conhecida como *two-grid* (Mazzotti *et al.*, 2016), onde existe uma grade de inversão e uma grade de modelagem. A primeira grade de malha mais esparsa, que possui baixa taxa de amostragem (poucos parâmetros), é onde os parâmetros são otimizados pelo o processo de inversão na busca por uma solução. A segunda grade é originada a partir da primeira, geralmente utilizando um procedimento de interpolação, e é utilizada no problema direto, ou seja, para gerar os dados sintéticos e possui uma alta taxa de amostragem (muitos elementos).

Para problemas 1D, a estratégia principal é conhecida como *block* ou *paramétrica*, onde os parâmetros do modelo são poucos, o mínimo possível sem perda na precisão da descrição do modelo de interesse, e são usados para gerar um modelo de aplicação, ou seja, compatível com a modelagem direta. Em geral, podemos definir um operador de parametrização Γ que converte um conjunto de parâmetros $\mathbf{m} = (m_1, m_1, \dots, m_n)$ em um vetor ou matriz que representa um escalar do campo de velocidade na equação da onda. Este operador pode ser definido, matematicamente, por:

$$\mathbf{c} = \Gamma(\mathbf{m}), \quad (2.9)$$

onde Γ é matematicamente simples de definir. Desta forma, o problema pode ser apresentado assim:

$$\begin{aligned} &\text{Minimizar } \phi(\mathbf{m}) \\ &\text{Sujeito a } \begin{cases} \mathbf{G}\mathbf{c} = \mathbf{d}^* \\ L_{inf_j} \leq m_j \leq L_{sup_j}, \quad j = 1, 2, \dots, n. \end{cases} \end{aligned} \quad (2.10)$$

2.3 Modelagem do problema sísmico 1D

A modelagem é um processo direto, que pode ser usado para simular a propagação do campo de onda, com a finalidade de reproduzir a resposta do modelo geológico em

subsuperfície. Esse processo, na forma inversa, permite estimar como seriam os dados sísmicos obtidos em diferentes profundidades de registro, a partir de medidas registradas na superfície. Na forma direta, permite sintetizar dados sísmicos, registrados na superfície, a partir de um modelo geológico fornecido.

2.3.1 Equação da onda

A modelagem sísmica direta corresponde ao modelo matemático da propagação da onda sísmica no meio geológico. O operador matemático G é representado por um conjunto de Equações Diferenciais Parciais (EDP). Neste estudo, o fenômeno é regido pela equação da onda isotrópica em uma dimensão (1D), dada a partir da seguinte expressão:

$$\left[\frac{1}{c^2(x)} \frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial x^2} \right] u(x, t) = f(t) \delta(x - x_s), \quad (2.11)$$

onde c representa a velocidade de propagação da onda; u , o campo de amplitude ou campo de pressão; $\delta(x - x_s)$, a “função” delta de Dirac com a posição da fonte x_s ; $f(t)$, a assinatura da fonte sísmica; e t , o tempo de duração da propagação da onda (Reynolds, 1978).

A assinatura da fonte sísmica utilizada neste trabalho é representada pela wavelet chapéu mexicano, também chamada de wavelet de Ricker, que é muito usual na literatura, obtida a partir da segunda derivada de uma função gaussiana (Ricker, 1940). A expressão matemática da fonte de Ricker no domínio do tempo é da forma:

$$f(t) = \left[1 - 2\pi^2 \nu_0^2 (t - t_0)^2 \right] e^{-\pi^2 \nu_0^2 (t - t_0)^2}, \quad (2.12)$$

sendo $t_0 = \frac{6}{\pi \nu_0 \sqrt{2}}$. Onde a variável t é janela do tempo em segundos para uma determinada

frequência de pico ν_0 em Hertz e $(t - t_0)$ é o tempo defasado.

2.3.1.1 Condições iniciais e de contorno

Para esse tipo de EDP, Equação (2.11), é necessário incluir duas condições iniciais:

$$\begin{cases} u(x,t) = 0 \\ \frac{\partial}{\partial t} u(x,t) = 0 \end{cases} \quad t \leq 0 \quad (2.13)$$

e para evitar reflexões nas bordas são inseridas duas condições de contorno dadas por:

$$\begin{cases} \frac{1}{c(x_0)} \frac{\partial}{\partial t} u(x_0,t) - \frac{\partial}{\partial x} u(x_0,t) = 0 \\ \frac{1}{c(x_L)} \frac{\partial}{\partial t} u(x_L,t) - \frac{\partial}{\partial x} u(x_L,t) = 0 \end{cases}, \quad (2.14)$$

onde x_0 e x_L são os limites das bordas do domínio físico de interesse.

As condições de contorno são um artifício matemático, introduzido na modelagem, para impedir que o campo de onda seja refletido nas fronteiras do modelo, retornando e interferindo no resultado final. A resolução numérica da Equação (2.14), no domínio computacional, é possível devido a implementação das Condições de Bordas de Absorção (Reynolds, 1978).

2.3.2 Solução numérica da equação da onda

O problema direto sísmico, ou modelagem direta, consiste na determinação da propagação do campo de onda no meio geológico mediante um modelo matemático, representado pelas Equações (2.11) à (2.14), que descrevem o fenômeno físico. Com isso, torna-se possível prever se os dados observados, registrados em um sismograma, correspondem a um determinado modelo. Para a onda acústica, esse modelo é composto pela velocidade de propagação da onda sísmica para cada ponto do meio.

A modelagem direta pode ser resolvida numericamente pelo Método dos Elementos Finitos (MEF), Método das Diferenças Finitas (MDF), entre outros. Na presente pesquisa, optou-se por utilizar o MDF, discretizando o domínio contínuo sobre uma malha uniforme de

nx pontos nodais igualmente espaçados Δx . Em cada um desses pontos, serão obtidas soluções aproximadas através da substituição das derivadas por diferenças finitas. Ou seja, a solução numérica da EDP pelo MDF tem como finalidade utilizar apenas os valores de $U_i^t \approx u(x, t)$, transformando o problema contínuo em um problema discreto que possa ser resolvido computacionalmente.

A Figura 2.2 mostra o esquema de discretização no domínio espacial, o estêncil utilizado, a região onde se aplica a equação que rege o fenômeno (pontos verdes) e as condições de bordas CB. Tal discretização das derivadas é realizada a partir da série de *Taylor*, que é a base para todas as aproximações numéricas usadas pelo MDF. O estêncil contém a lista de pontos nodais usados no esquema de aproximação por diferenças finitas, incluindo o ponto em análise.

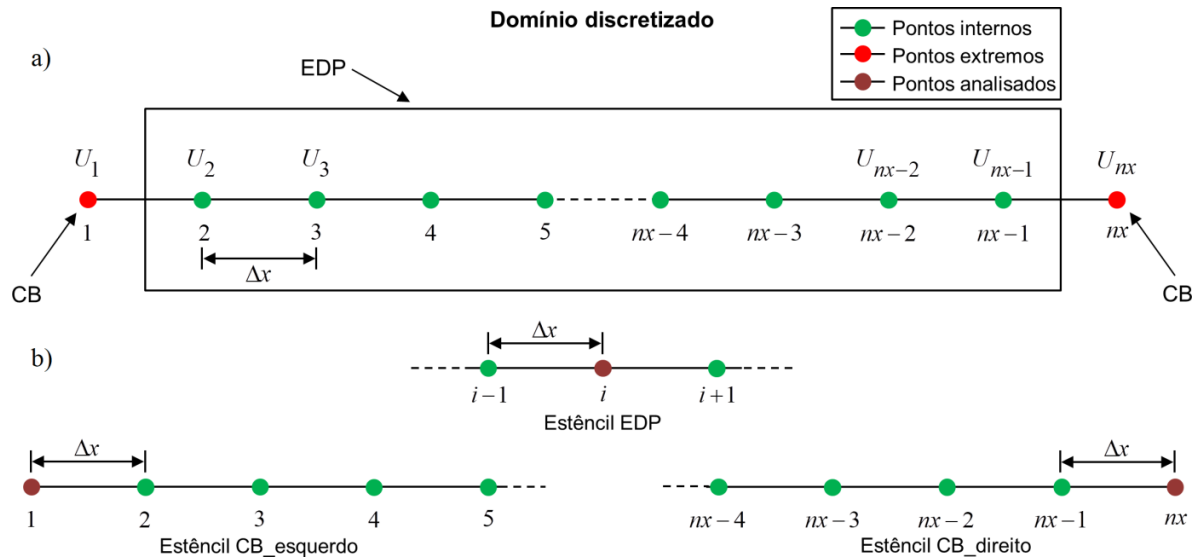


Figura 2.2 – (a) Malha uniforme de nx pontos 1D; (b) Estêncil

Neste trabalho, assim como na maioria das aplicações do MDF para a Equação (2.11), as derivadas segunda temporais e espaciais foram aproximadas por fórmulas de diferenças finitas centradas com precisão de segunda ordem, usando um esquema padrão de três pontos (estêncil) em 1D (Alford *et al.*, 1974), resultando na Equação (2.15). Os pontos internos na Figura 2.2 representa o domínio discreto para aplicação dessa equação.

$$\frac{1}{c_i^2} \frac{U_i^{t-1} - 2U_i^t + U_i^{t+1}}{(\Delta t)^2} - \frac{U_{i-1}^t - 2U_i^t + U_{i+1}^t}{(\Delta x)^2} = f_i^t \quad (2.15)$$

Existem muitos métodos para resolver a Equação (2.14), de acordo com (Gao *et al.*, 2017), uma das soluções mais precisas e simples é conhecida como limite de Reynolds. Nesse método, as aproximações para as derivadas primeira são realizadas utilizando fórmulas de diferenças finitas com precisão de primeira ordem, avançadas no tempo para ambos os limites das bordas; e avançadas e atrasadas no espaço para os limites das bordas esquerda e direita, respectivamente (Reynolds, 1978). Porém, com o intuito de melhorar os resultados numéricos do limite de Reynolds original, o método foi modificado para esta pesquisa, aplicando as fórmulas de diferenças finitas com precisão de quarta ordem espacial para as bordas, resultando nas Equações (2.16). Os pontos extremos na Figura 2.2 mostram o domínio discreto para aplicação dessa equação.

$$\text{CB} \begin{cases} \frac{1}{c_1} \frac{U_1^{t+1} - U_1^t}{\Delta t} - \frac{(-25/12)U_1^{t+1} + 4U_2^t - 3U_3^t + (4/3)U_4^t - (1/4)U_5^t}{\Delta x} = 0 \\ \frac{1}{c_{nx}} \frac{U_{nx}^{t+1} - U_{nx}^t}{\Delta t} + \frac{(1/4)U_{nx-4}^t - (4/3)U_{nx-3}^t + 3U_{nx-2}^t - 4U_{nx-1}^t + (25/12)U_{nx}^t}{\Delta x} = 0 \end{cases} \quad (2.16)$$

Isolando o campo de amplitude no tempo futuro U^{t+1} na Equação (2.15) em função dos campos no tempo presente U^t e passado U^{t-1} , obtém-se o seguinte esquema explícito para o MDF no domínio numérico interno,

$$U_i^{t+1} = \lambda_i^2 U_{i-1}^t + (2 - 2\lambda_i^2) U_i^t + \lambda_i^2 U_{i+1}^t - U_i^{t-1} + \xi_i^2 f_i^t, \text{ com } i = 2, \dots, nx-1 \quad (2.17)$$

onde os coeficientes são dados por:

$$\text{Com} \begin{cases} \lambda_i = c_i \frac{\Delta t}{\Delta x}, \\ \xi_i = c_i \Delta t \end{cases} \quad (2.18)$$

sendo Δx e Δt , respectivamente, o passo do espaço e do tempo no processo de discretização.

$$e \ f_i^t = \begin{cases} f(t), & \text{se } x = x_s \\ 0, & \text{se } x \neq x_s \end{cases} \quad (2.19)$$

Aplicando o mesmo procedimento nas Equações (2.16), obtém-se as expressões do campo de amplitudes U^{t+1} em função apenas do tempo presente U^t , como segue:

$$U_1^{t+1} = \left(1 - \frac{25}{12}\lambda_1\right)U_1^t + 4\lambda_1 U_2^t - 3\lambda_1 U_3^t + \frac{4}{3}\lambda_1 U_4^t - \frac{1}{4}\lambda_1 U_5^t \quad (2.20)$$

$$U_{nx}^{t+1} = -\frac{1}{4}\lambda_{nx} U_{nx-4}^t + \frac{4}{3}\lambda_{nx} U_{nx-3}^t - 3\lambda_{nx} U_{nx-2}^t + 4\lambda_{nx} U_{nx-1}^t + \left(1 - \frac{25}{12}\lambda_{nx}\right)U_{nx}^t \quad (2.21)$$

Esses esquemas de diferenças finitas, Equações (2.17), (2.20) e (2.21), podem ser escritos como um sistema linear ou como uma equação matricial, como segue:

$$\mathbf{U}^{t+1} = \mathbf{A}\mathbf{U}^t - \mathbf{U}^{t-1} + \mathbf{F}^t \quad (2.22)$$

$$\begin{bmatrix} U_1^{t+1} \\ U_2^{t+1} \\ \vdots \\ U_{nx-1}^{t+1} \\ U_{nx}^{t+1} \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} U_1^t \\ U_2^t \\ \vdots \\ U_{nx-1}^t \\ U_{nx}^t \end{bmatrix} - \begin{bmatrix} 0 \\ U_2^{t-1} \\ \vdots \\ U_{nx-1}^{t-1} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \xi_2^2 f_2^t \\ \vdots \\ \xi_{nx-1}^2 f_{nx-1}^t \\ 0 \end{bmatrix} \quad (2.23)$$

onde a matriz \mathbf{A} de tamanho $(nx \times nx)$, é representada da seguinte forma:

$$\mathbf{A} = \begin{bmatrix} 1 - \frac{25}{12}\lambda_1 & 4\lambda_1 & -3\lambda_1 & \frac{4}{3}\lambda_1 & -\frac{1}{4}\lambda_1 & \cdots & 0 \\ \lambda_2^2 & 2 - 2\lambda_2^2 & \lambda_2^2 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_3^2 & 2 - 2\lambda_3^2 & \lambda_3^2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \lambda_{nx-2}^2 & 2 - 2\lambda_{nx-2}^2 & \lambda_{nx-2}^2 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{nx-1}^2 & 2 - 2\lambda_{nx-1}^2 & \lambda_{nx-1}^2 \\ 0 & 0 & -\frac{1}{4}\lambda_{nx} & \frac{4}{3}\lambda_{nx} & -3\lambda_{nx} & 4\lambda_{nx} & 1 - \frac{25}{12}\lambda_{nx} \end{bmatrix} \quad (2.24)$$

A equação matricial representa um esquema de evolução da variável temporal para os pontos extremos e internos, onde a solução U_i^t é encontrada de forma explícita para cada passo de tempo. Além disso, a condição de Courant-Friedrichs-Lewy (CFL) é um critério de estabilidade temporal que deve ser satisfeito para que o sistema possa convergir para uma solução precisa. Essa condição impõe restrições no tamanho do intervalo de tempo permissível, que é dado matematicamente pela Equação (2.25) (Alford *et al.*, 1974).

$$\Delta t \leq \alpha \frac{\Delta x}{c_{max}} \quad (2.25)$$

onde $c_{max} = \|\mathbf{c}\|_{\infty}$ é a velocidade máxima de propagação da onda no meio geológico (modelo de velocidade) e Δx é o incremento espacial da malha uniforme. Para uma solução estável, devido às aproximações usadas neste trabalho, $\alpha = \frac{1}{6}$

2.4 Otimização livre de derivadas

Meta-heurísticas são procedimentos DFO finitos, passo a passo e bem definidos, que são capazes de resolver um problema de otimização. Ao contrário das heurísticas, elas são dotadas de mecanismos mais sofisticados e fazem uso de diferentes estratégias para explorar e explorar, de forma mais abrangente e eficiente, o espaço de busca de uma solução ótima. A maioria das meta-heurísticas é inspirada em fenômenos da natureza; provida de mecanismos para fuga de ótimos locais. Alguns exemplos de algoritmos bioinspirados (meta-heurísticas bioinspiradas), ou seja, inspirados na biologia são: Algoritmo Genético (GA), inspirado na teoria da evolução genética de Darwin (Mitchell, 1996), e Otimização por Enxame de Partículas (PSO), inspirada no comportamento de um bando de pássaros em voo interagindo uns com os outros (Kennedy, 1995; Kennedy, 2001). Enquanto outros são inspirados em fenômenos físico, como Simulated Annealing (SA) (Kirkpatrick *et al.*, 1983).

As técnicas de DFO não bioinspiradas podem ser puramente estocásticas, como o método de Monte Carlo (MC) (Metropolis e Ulam, 1949). Existem várias abordagens baseadas em operações matemáticas fundamentais que permite as transformações geométricas do simplex (poliedro convexo), e uma delas é o método Nelder-Mead Simplex (Nelder e

Mead, 1965; Lagarias *et al.*, 1998). Além disso, existem métodos que combinam as duas características, estocástica e evolucionária, como o método Controlled Random Search (CRS) (Price, 1983).

A maior parte do conhecimento sobre esses métodos é de natureza empírica devido à dificuldade em fazer estudos analíticos, impossibilitando uma análise de convergência rigorosa para o ótimo global. Além disso, são relativamente lentos e costuma ser mal dimensionado com o número de variáveis de decisão do problema. No entanto, esses métodos são robustos e fáceis de implementar, funcionando bem com funções objetivo complicadas, como é o caso do problema FWI. Conforme Koduru *et al.* (2006), as principais vantagens das técnicas bioinspiradas de otimização estocástica (ou meta-heurísticas) são: livres de derivadas, não ficam presas facilmente em mínimos locais, amostram uma ampla região do espaço de busca, podem ser personalizadas para atender a um problema específico e podem ser facilmente hibridizadas com outros algoritmos para melhorar o seu desempenho geral.

Como mencionado anteriormente, PSO e Nelder-Mead Simplex e suas respectivas variantes são os algoritmos mais importantes no contexto atual, motivos pelos quais foram integrados para compor o algoritmo híbrido desse trabalho. Portanto, eles são descritos com mais detalhes a seguir. Além desses dois também apresentamos o algoritmo K-means desenvolvido por Hartigan and Wong, (1979), empregado para explorar melhor o espaço de busca.

2.4.1 Otimização por enxame de partículas

O algoritmo de Otimização por Enxame de Partículas (PSO) é um método de otimização estocástico baseado em população, desenvolvido pelo psicólogo social James Kennedy e pelo engenheiro eletricista Russel Eberhart em 1995 (Kennedy e Eberhart, 1995). Esse método é inspirado no comportamento coletivo de um sistema biológico social auto-organizado e descentralizado, como por exemplo, a coreografia imprevisível de um bando de pássaros ou de um cardume de peixes interagindo entre si e com o meio ambiente em busca do caminho para um local que contenha alimento (Kennedy, 1997).

O algoritmo PSO é baseado em inteligência de enxames, no qual a população é inicializada com um conjunto de partículas (soluções candidatas), cada uma com uma posição e uma velocidade aleatórias no espaço de busca de n dimensões. O algoritmo considera um espaço de busca viável, ou seja, limitado pelas restrições do problema, mantendo um enxame de p partículas. A posição de cada partícula i na iteração k , é representada pelo o vetor $\mathbf{X}_i^k = (x_{i1}^k, x_{i2}^k, \dots, x_{in}^k)$.

Ao final de cada iteração do algoritmo, as posições de todas as partículas são modificadas adicionando-se a cada uma delas um vetor de velocidade, $\mathbf{V}_i = (v_{i1}, v_{i2}, \dots, v_{in})$. Assim, as atualizações das posições das partículas são realizadas usando a seguinte equação:

$$\mathbf{X}_i^{k+1} = \mathbf{X}_i^k + \mathbf{V}_i^{k+1} \quad (2.26)$$

O processo de otimização é impulsionado pelo vetor de velocidade, que é atualizado a cada iteração, permitindo que as partícula se movam em direção a um local melhor no espaço de busca. Para isso, são considerados: a velocidade anterior, a melhor posição em que a i —ésima partícula já esteve, e a melhor localização já ocupada por qualquer partícula do enxame até a iteração atual. Assim, a atualização da velocidade no PSO clássico é determinada pela seguinte Equação (2.27), (Koduru *et al.*, 2006):

$$\mathbf{V}_i^{k+1} = \mathbf{V}_i^k + C_1 r_1 (\mathbf{X}_{ib}^k - \mathbf{X}_i^k) + C_2 r_2 (\mathbf{X}_{gb}^k - \mathbf{X}_i^k) \quad (2.27)$$

com $\mathbf{X}_i^k \in \Omega \subset \mathbb{R}^n$, onde Ω é o espaço de busca viável que respeita todas as restrições, definido em função do produto cartesiano de n intervalos fechados:

$$\Omega = [L_{inf_1}, L_{sup_1}] \times [L_{inf_2}, L_{sup_2}] \times \dots \times [L_{inf_n}, L_{sup_n}] \quad (2.28)$$

sendo L_{inf_j} e L_{sup_j} , denominados, respectivamente, limites inferior e superior do espaço de busca correspondente a uma hipercaixa Ω para todo $j = 1, \dots, n$.

Na Equação (2.27), C_1 e C_2 são chamados de coeficientes de aceleração ou parâmetros cognitivo e social, respectivamente; r_1 e r_2 são os valores aleatórios uniformemente distribuídos no intervalo $[0, 1]$; \mathbf{X}_{ib}^k é a melhor posição já encontrada pela i – ésima partícula até o momento, ou seja, o melhor individual; \mathbf{X}_{gb}^k é a melhor posição encontrada pelo enxame até a iteração atual, ou seja, o melhor global.

O parâmetro cognitivo quantifica o desempenho individual de cada partícula, fazendo com que cada uma seja atraída de volta à sua melhor posição, sendo denominado de termo de memória individual (nostalgia). O parâmetro social quantifica o desempenho de cada partícula coletivamente em uma determinada vizinhança, fazendo com que todas as partículas sejam atraídas para a melhor posição determinada por aquela vizinhança, sendo chamado de termo de memória coletiva. Dessa forma, o movimento das partículas ocorre por meio de memória e cooperação.

O PSO usa uma função objetivo $f(\mathbf{X}_i^k)$, a fim de avaliar cada solução \mathbf{X}_i^k , na k – ésima iteração. Neste contexto, o valor de aptidão de cada partícula i do enxame é $\tilde{fit}(\mathbf{X}_i^k) = f(\mathbf{X}_i^k)$, onde $\tilde{fit}(\bullet)$ é a forma generalizada para qualquer função objetivo. No caso do FWI, $\tilde{fit}(\bullet) = \phi(m)$.

Para uma função $f(\mathbf{X}): \Re^n \rightarrow \Re$ considerando um problema de minimização e uma vizinhança global, as etapas do algoritmo PSO para cada iteração k podem ser listadas da seguinte forma, (Shi e Eberhart, 1998):

0. Inicialização:

Inicializar aleatoriamente todas as posições \mathbf{X}_i^k e velocidades \mathbf{V}_i^k , para o enxame de p partículas de dimensão n , e definir $k = 1$;

1. Melhores posições:

Para cada partícula i do enxame **faça**

Calcular o valor de aptidão $fit(\mathbf{X}_i^k) = f(\mathbf{X}_i^k)$;

Se o valor de aptidão atual, $fit(\mathbf{X}_i^k)$, é menor que o do melhor individual $f(\mathbf{X}_{ib}^k)$, ou seja, $fit(\mathbf{X}_i^k) < f(\mathbf{X}_{ib}^k)$, então

Atualizar o valor do melhor individual \mathbf{X}_{kb}^k para receber o valor da partícula atual, $\Rightarrow \mathbf{X}_{ib}^k = \mathbf{X}_i^k$;

Fim

Se o valor de aptidão atual, $fit(\mathbf{X}_{ib}^k)$, é menor que o do melhor global $f(\mathbf{X}_{gb}^k)$, ou seja, $fit(\mathbf{X}_{ib}^k) < f(\mathbf{X}_{gb}^k)$ então

Atualizar o valor do melhor global \mathbf{X}_{gb}^k para receber o valor da partícula atual, $\Rightarrow \mathbf{X}_{gb}^k = \mathbf{X}_{ib}^k$;

Fim

2. Atualização do enxame:

Para todo o enxame **faça**

Atualizar o vetor de velocidade,

$$\mathbf{V}_i^{k+1} = \mathbf{V}_i^k + C_1 r_1 (\mathbf{X}_{ib}^k - \mathbf{X}_i^k) + C_2 r_2 (\mathbf{X}_{gb}^k - \mathbf{X}_i^k)$$

Atualizar o vetor de posição,

$$\mathbf{X}_i^{k+1} = \mathbf{X}_i^k + \mathbf{V}_i^{k+1}$$

3. Faça $k = k+1$ e,

Enquanto um critério de parada não for satisfeito, volte ao passo 1 para iniciar uma nova iteração.

O critério de parada usual do algoritmo PSO é o número máximo de iterações $(k > It_{max})$.

De acordo com Rana *et al.* (2010), as principais vantagens do algoritmo PSO são os poucos parâmetros a serem otimizados e a necessidade de pouca memória de execução. No entanto, existem desvantagens que limitam sua aplicação, como a taxa de convergência muito lenta quando próximo da solução global e a convergência rápida e prematura em pontos não ótimos. O problema da convergência lenta está relacionado às partículas que convergem para o melhor global, quando todas as partículas se movem em direção a um único ponto entre a melhor posição global e a melhor posição individual. Além disso, o rápido fluxo de informações entre as partículas é outro motivo para esse problema, resultando na criação de

partículas semelhantes (perda de diversidade), o que aumenta o risco de prendê-las em um mínimo local. Por outro lado, a convergência prematura em funções multimodais está relacionado à fraca capacidade de busca local do algoritmo PSO (Fakhouri *et al.*, 2020).

Podemos dividir os parâmetros que caracterizam o método PSO em duas classes: parâmetros iniciais e de controle. Os parâmetros iniciais são: o tamanho do enxame M , as posições das partículas \mathbf{X}_i^k , as velocidades \mathbf{V}_i^k e o número máximo de iterações permitidas It_{max} . Os parâmetros de controle são: os coeficientes de aceleração C_1 e C_2 , a velocidade máxima limitada V_{max} , a topologia do enxame (vizinhança) e o critério de parada. Além do It_{max} o critério de parada pode ser baseado na precisão da atualização da melhor solução. Em geral, todos esses parâmetros são definidos pelo usuário e precisam ser escolhidos com cuidado. Existe na literatura científica uma quantidade significativa de pesquisas que ajudam a entendê-los e defini-los. A seguir, apresentamos algumas considerações e análises dos valores admissíveis para tais parâmetros.

Os coeficientes de aceleração C_1 e C_2 determinam a tendência da busca. Se $C_1 = C_2 = 0$, então as partículas movem-se na mesma direção até o limite do espaço de busca; se $C_1 > 0$ e $C_2 = 0$, então cada partícula realiza sua busca local, sendo atraídas para \mathbf{X}_{ib}^k ; se $C_1 = 0$ e $C_2 > 0$, todas as partículas são atraídas por um único ponto \mathbf{X}_{gb}^k ; se $C_1 = C_2 > 0$, cada partícula é atraída para o baricentro entre \mathbf{X}_{ib}^k e \mathbf{X}_{gb}^k ; se $C_1 \gg C_2$, as partículas são atraídas para suas posições \mathbf{X}_{ib}^k , resultando em dispersão; e se $C_1 \ll C_2$, as partículas são atraídas para \mathbf{X}_{gb}^k , resultando em convergência prematura para ótimos locais. Com valores baixos de C_1 e C_2 , o resultado é um movimento suave das partículas. Com valores altos desses coeficientes, os movimentos das partículas são abruptos.

2.4.2 Otimização por enxame de partículas modificado

O algoritmo PSO clássico foi modificado por (Shi e Eberhart, 1998), incluindo um novo coeficiente, chamado de fator de inércia, com o objetivo de aprimorar a velocidade de convergência. Assim, tem-se uma nova proposta de atualização da velocidade apresentada conforme a Equação (2.29):

$$\mathbf{V}_i^{k+1} = w\mathbf{V}_i^k + C_1r_1(\mathbf{X}_{ib}^k - \mathbf{X}_i^k) + C_2r_2(\mathbf{X}_{gb}^k - \mathbf{X}_i^k) \quad (2.29)$$

onde w é o fator de inércia, que ajuda a manter a estabilidade da convergência. Um coeficiente inercial maior geralmente torna a exploração global mais fácil, ou seja, favorece a busca por novas áreas dentro do espaço de busca, enquanto um coeficiente menor facilita a exploração local (Fakhouri *et al.*, 2020).

A Figura 2.3(a) mostra as operações vetoriais para a atualização da posição de uma partícula i do enxame na k – ésima iteração. Nessa Figura a partícula tem 2 dimensões, $n=2$, ou seja, $\mathbf{X}_i^k = (x_{i1}^k, x_{i2}^k)$. Tem-se então: o termo de inércia, que força a partícula a se mover na mesma direção; o termo cognitivo, que força a partícula a seguir sua melhor posição; e o termo de aprendizagem social, que força a partícula a seguir a direção da melhor posição do enxame.

Um dos grandes desafios das metaheurísticas baseadas em um enxame de partículas é encontrar um equilíbrio ideal entre os mecanismos de exploração (diversificação) e exploração (intensificação). A exploração proporciona uma busca maior por novas soluções, identificando regiões com potencial para melhores soluções. Por outro lado, a exploração proporciona uma busca mais focada a partir das soluções já encontradas, buscando assim a melhor solução em uma região. A Figura 2.3(b) ilustra o equilíbrio entre esses mecanismos de busca por melhores soluções.

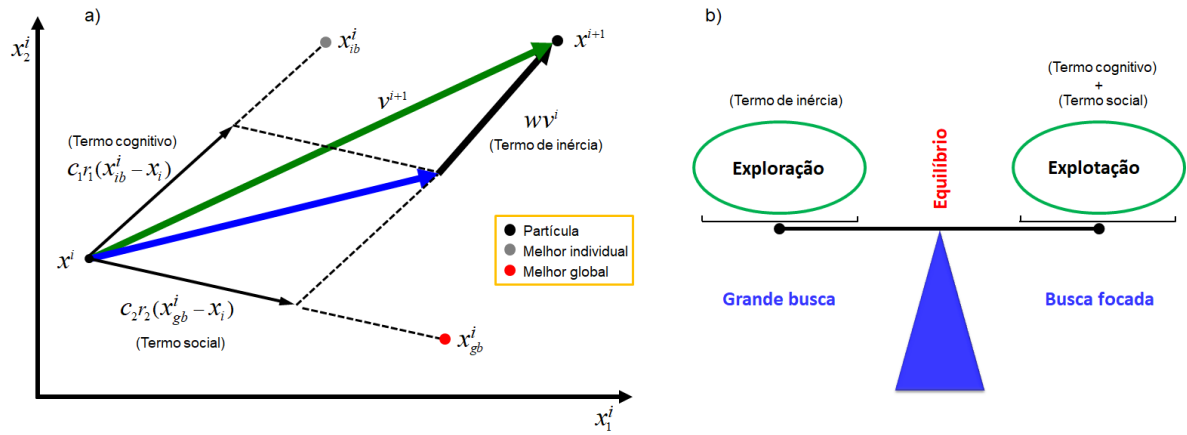


Figura 2.3 – (a) Os três vetores associados à atualização da velocidade da partícula no algoritmo PSO original; (b) Equilíbrio entre os mecanismos de exploração e explotação.

O termo inercial w controla o impacto da velocidade atual na atualização da direção da partícula. Como mencionado anteriormente, um valor maior do termo de inércia contempla a exploração (busca global), tornando o enxame divergente, enquanto um valor menor do termo de inércia contempla a explotação (busca local), desacelerando as partículas. Existem várias possibilidades para o valor de w : uma constante, um valor multiplicado por uma taxa de amortecimento em cada iteração e uma variável reduzida linearmente com as iterações, entre outras opções (Clerc e Kennedy, 2002).

Clerc, (1999) propôs um limitador para a velocidade inercial denominado de fator de constrição, que delimita os coeficientes de aceleração, necessário para garantir a convergência do algoritmo. A Equação (2.30) mostra como esse fator é calculado:

$$K = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}, \text{ onde } \varphi = C_1 + C_2, \varphi > 4 \quad (2.30)$$

Assim, a atualização da velocidade passa a ser definida conforme a Equação:

$$\mathbf{V}_i^{k+1} = K \left[\mathbf{V}_i^k + C_1 r_1 (\mathbf{X}_{ib}^k - \mathbf{X}_i^k) + C_2 r_2 (\mathbf{X}_{gb}^k - \mathbf{X}_i^k) \right] \quad (2.31)$$

Em todos os casos para os quais o método de Clerc foi usado na pesquisa realizada por (Eberhart e Shi, 2000), a soma de C_1 e C_2 foi definida como 4,1. Isso resulta em $K = 0,729$ e $C_1 = C_2 = 2,05$. Em consequência disso, a velocidade anterior e cada um dos coeficientes de

aceleração foram multiplicadas por 0,729. Isso foi observado e notado que é equivalente a usar a Equação (2.29) com $w = 0,729$ e $C_1 = C_2 = 1,49445$. (Shi e Eberhart, 1998) descobriram que quando $w \in (0,8; 1,2)$ o PSO terá melhor chance de encontrar o ótimo global. Porém, com $w \in [0,9; 1,2]$ o algoritmo terá em média um melhor desempenho com um número razoável de iterações para encontrar o ótimo global. Além disso, foi introduzido um fator de inércia decrescente ao longo do tempo, que trouxe uma melhoria significativa no desempenho do PSO. (Eberhart e Shi, 2001) sugeriram $C_1 = C_2 = 2$ e $w = 0,5 + \frac{rand}{2}$. Outros experimentos (Shi e Eberhart 1998; Shi e Eberhart, 1999) mostraram que ao inserir um fator de inércia linearmente decrescente de 0,9 para 0,4 durante as primeiras iterações, o desempenho do algoritmo PSO ficou bastante aprimorado fornecendo bons resultados. Esses estudos mostraram que, com o uso de uma estratégia de peso inercial adaptativo, o PSO tem a capacidade de convergir rapidamente, seu desempenho torna-se insensível ao tamanho da população e o método é bem dimensionado. De acordo com (Dai *et al.*, 2011), quando $w \in [0,5; 0,75]$, a taxa de sucesso para o ótimo é muito alta, mais de 80%.

Uma versão do PSO com coeficientes de aceleração variáveis no tempo foi considerada em (Ratnaweera *et al.*, 2004). Os melhores intervalos sugeridos para C_1 e C_2 na maioria das funções de benchmark foram $[2,5; 0,5]$ e $[0,5; 2,5]$, respectivamente. Esses valores foram gradualmente variados ao longo das iterações. O procedimento tendeu a aumentar a diversidade no início busca, permitindo o alcance de todo o espaço de busca. Além disso, quando o algoritmo estava convergindo para a solução ótima, tendeu a intensificar o ajuste fino das soluções durante a última parte da busca para encontrar o ótimo global de forma eficiente. (Suganthan, 1999) usou estratégia de ajuste gradual dos parâmetros para os pesos inerciais e os coeficientes de aceleração. Essa estratégia foi realizada com o objetivo de fazer um ajuste fino durante os estágios finais da otimização.

Como a otimização adaptativa por enxame de partículas (APSO) fornece, em geral, melhor eficiência de busca do que o PSO padrão, então o coeficiente de inércia, w , os coeficientes de aceleração, C_1 e C_2 , entre outros parâmetros, foram considerados dinâmicos no presente trabalho, com variação linear ao longo das iterações. Os parâmetros e coeficientes

acima mencionados são atualizados por meio da Equação (2.32), na qual S identifica cada um deles.

$$S = S_{inicial} - \left(\frac{It}{It_{max}} \right) (S_{inicial} - S_{final}) \quad (2.32)$$

A ideia é que, inicialmente, as experiências individuais de cada partícula recebam maior importância e, aos poucos, a vivência coletiva seja favorecida. Isso permite que o enxame explore melhor o espaço de busca de soluções, dando maior diversidade ao algoritmo, evitando assim uma possível convergência prematura. Além disso, foram adotados limites para as velocidades das partículas. A fixação da velocidade é calculada como uma porcentagem η , do intervalo do espaço de busca ao longo dos eixos de coordenados, pela seguinte Equação (2.33), (Evers e Ghalia *et al.*, 2004):

$$V_{j,max} = \eta \times range_j(\Omega) \quad (2.33)$$

onde

$$range_j(\Omega) = L_{sup_j} - L_{inf_j}, \quad j = 1, \dots, N \quad (2.34)$$

Então, faz-se $V^{k+1} \in [-V_{j,max}, V_{j,max}]$. A porcentagem de fixação da velocidade é geralmente escolhida entre $0,1 \leq \eta \leq 0,5$. Nesse trabalho foi adotado $\eta = 1,5$.

Quanto à influência do tamanho da população Dai, Liu e Li (Dai *et al.*, 2011) concluíram que, quando o número de partículas é muito pequeno, a iteração é rápida, mas a taxa de convergência é baixa e o desempenho da busca global é pobre. Quando o número de partículas aumentou para 10, houve um ganho de 80%. Quando o número de partículas aumentou em mais de 20, não houve melhora e aumentou o tempo de CPU. Os estudos feitos por Shi e Eberhart (Shi e Eberhart, 1998) indicaram que o PSO não é sensível ao tamanho da população. No entanto, nos estudos realizados por Carlisle e Dozier (Carlisle e Dozier, 2001), foi descoberto que isso geralmente é verdade em termos de desempenho, mas não em termos

de custo computacional. Além disso, em estudos com funções de benchmark para otimização, foi concluído que uma população com 30 partículas parece ser uma boa escolha.

A topologia do enxame define a vizinhança de cada partícula, onde ocorre a comunicação entre elas. Isso envolve o número de partículas vizinhas que influenciam o movimento de uma determinada partícula do enxame. Diferentes topologias têm sido usadas para controlar o fluxo de informações entre as partículas. Kennedy (Kennedy, 1999) e Medina *et al.* (Medina *et al.*, 2009) projetaram quatro topologias diferentes, incluindo círculo, roda, estrela e arestas aleatórias. O PSO padrão usa a topologia em estrela como estrutura de comunicação. Nesta abordagem, todas as partículas se comunicam globalmente, compartilhando a melhor posição de uma única partícula. Isso pode causar uma convergência prematura do enxame. Portanto, é necessário pesquisar a vizinhança dos indivíduos para melhorar o desempenho do PSO ao longo das iterações. Por exemplo, a vizinhança pode ser gradualmente expandida de uma partícula individual para incluir todas as partículas do enxame. Nesta pesquisa não é considerada a análise da vizinhança.

Em Eberhart e Kennedy (Eberhart e Kennedy, 1995), os autores concluíram que uma vizinhança local é melhor para evitar mínimos locais, enquanto uma vizinhança global converge mais rápido. Em um experimento com 30 partículas, Carlisle e Dozier (Carlisle e Dozier, 2001) variaram o tamanho da vizinhança de 2 para o global em etapas de 2 e chegaram à seguinte conclusão: uma vizinhança global parece ser uma escolha geral melhor, pois parece alcançar os mesmos resultados com menos trabalho. Suganthan (Suganthan, 1999), em um experimento que gradualmente aumentou o tamanho da vizinhança para o global, obteve uma análise inconclusiva dos resultados. Em Wang *et al.* (Wang *et al.*, 2013), um algoritmo PSO híbrido foi proposto. Ele empregou um mecanismo de aumento da diversidade e estratégias de busca reversa na vizinhança para alcançar um compromisso entre as habilidades de exploração e exploração.

Neste trabalho, foi adotado a estratégia de topologia global. Além disso, foi seguindo as ideias contidas nas pesquisas de Riget e Vesterstrøm (Riget e Vesterstrøm, 2002) e de Wang (Wang *et al.*, 2013), sobre otimização por enxame de partículas atrativas e repulsivas. O número de partículas repulsivas é dinâmico, diminuindo linearmente com as iterações. Além disso, essas partículas repulsivas são rebeldes, ou seja, ora são repulsivas, ora são

atrativas, dependendo se um determinado limite τ é atingido. O limite também decai com as iterações. O número de partículas repulsivas (possivelmente rebeldes) é uma porcentagem rr_p da população do enxame. Novamente, a ideia é adicionar maior diversidade ao algoritmo PSO. No início, o enxame tem um comportamento mais adverso e tende a um comportamento mais colaborativo no final. A atualização das velocidades das partículas é dada pela Equação (2.35).

$$\mathbf{V}^{k+1} = w\mathbf{V}^k + dir \left[C_1 r_1 (\mathbf{X}_{kb}^k - \mathbf{X}^k) + C_2 r_2 (\mathbf{X}_{gb}^k - \mathbf{X}^k) \right] \quad (2.35)$$

onde

$$dir = \begin{cases} -1, & \text{se } rand \leq \tau \\ 1, & \text{se } rand > \tau \end{cases} \quad (2.36)$$

O termo *dir* é usado para definir se as partículas rebeldes irão expandir ou contrair. Já o termo *rand* é um número aleatório no intervalo $[0, 1]$ obtido a cada iteração e $\tau \in [0, 1]$ é o limiar da rebelião, previamente definido.

2.4.3 Otimização com Simplex Adaptativo de Nelder-Mead

O método Simplex de Nelder-Mead publicado em 1965 por John Ashworth Nelder e Roger Mead, pertence à classe geral dos métodos determinísticos de busca direta que não utilizam derivadas. Ou seja, trata-se de um método de otimização não linear baseado em DFO.

O simplex é a figura geométrica do espaço n -dimensional contendo $n + 1$ vértices. No espaço bidimensional o simplex é um triângulo e no espaço tridimensional é um tetraedro.

O Simplex de Nelder-Mead é projetado para encontrar o ponto ótimo de uma função objetivo não linear e irrestrita em um espaço multidimensional. Basicamente, o Simplex de Nelder-Mead consiste em um método que minimiza uma função de n variáveis, avaliando esta função em $n + 1$ vértices. Nessa abordagem, o simplex é construído pela substituição do vértice de maior valor por outro de menor valor, ambos da função objetivo. O processo Simplex de Nelder-Mead é adaptativo, fazendo com que os simplex sejam continuamente

revisados para melhor se adequarem à natureza da superfície de resposta (topologia do espaço de busca). Ou seja, o próprio simplex se adapta à paisagem local e se contrai para uma solução local mínima. Do ponto de vista computacional, este método mostra-se ser compacto e eficaz para muitos problemas não lineares (Nelder e Mead, 1965; Olsson e Lloyd, 1975).

Quatro coeficientes escalares são necessários para o algoritmo Simplex de Nelder-Mead:

1. Coeficiente de reflexão: $\rho > 0$;
2. Coeficiente de expansão: $\chi > 1$ com $\chi > \rho$;
3. Coeficiente de contração: $0 < \gamma < 1$;
4. Coeficiente de redução: $0 < \sigma < 1$.

Na versão padrão do método Nelder-Mead simplex, esses coeficientes são fixos, (Nelder e Mead, 1965):

$$\rho = 1, \chi = 2, \gamma = 0,5 \text{ e } \sigma = 0,5.$$

A versão adaptada, (*Adaptive Nelder-Mead Simplex* - ANMS), é uma implementação do método Nelder-Mead em que esses coeficientes dependem da dimensão do problema de otimização, como segue (Gao e Han, 2010):

$$\rho = 1, \chi = 1 + \frac{2}{n}, \gamma = 0,75 - \frac{1}{2n} \text{ e } \sigma = 1 - \frac{1}{n}$$

Essa versão adaptada, ANMS, tenta superar as dificuldades de convergência encontradas pelo Simplex de Nelder-Mead para problemas com alta dimensão, $n > 2$. Observe que quando $n = 2$, o ANMS é idêntico ao Nelder-Mead Simplex padrão. Neste caso, os vértices do simplex formam um triângulo.

O método Nelder-Mead Simplex considera quatro operações distintas que movem o simplex em direção ao centroide dos n melhores vértices e uma operação que causa seu encolhimento em direção ao seu melhor vértice.

Sejam X_1, X_2, \dots, X_{n+1} os vértices que definem um simplex no \mathfrak{R}^n e os respectivos coeficientes de reflexão, expansão, contração e redução: ρ, χ, γ e σ . Para uma função objetivo $f(\mathbf{X}): \mathfrak{R}^n \rightarrow \mathfrak{R}$, os passos do algoritmo Simplex de Nelder-Mead para cada iteração k são os seguintes (Lagarias *et al.*, 1998):

0. **Receber:** os n pontos (vértices) de teste e defina $k = 1$;

1. **Ordenar:** os vértices do simplex de acordo com os valores da função objetivo:

$$f(X_1) \leq f(X_2) \leq \dots \leq f(X_{n+1}).$$

Calcule o centroide dos n melhores pontos:

$$C = \frac{1}{n} \sum_{j=1}^n X_j$$

2. **Reflexão:** calcular o ponto de reflexão X_r a partir:

$$X_r = (1 + \rho)C - \rho X_{n+1}$$

Se $f(X_1) \leq f(X_r) < f(X_n)$, aceitar o ponto X_r e ir para o passo 6.

3. **Expansão:** Se $f(X_r) < f(X_1)$, calcular o ponto de expansão X_e a partir:

$$X_e = (1 + \rho\chi)C - \rho\chi X_{n+1}$$

Se $f(X_e) < f(X_r)$, aceitar o ponto X_e e ir para o passo 6.

Caso contrário, $f(X_e) \geq f(X_r)$, aceitar o ponto X_r e ir para o passo 6.

4. **Contração:** Se $f(X_r) \geq f(X_n)$:

a) Se $f(X_n) \leq f(X_r) < f(X_{n+1})$, calcular o ponto de contração externa X_c a partir:

$$X_c = (1 + \rho\gamma)C - \rho\gamma X_{n+1}$$

Se $f(X_c) \leq f(X_r)$, aceitar o ponto X_c e ir para o passo 6.

Caso contrário, $f(X_c) > f(X_r)$, ir para o passo 5.

b) Se $f(X_r) \geq f(X_{n+1})$, calcular o ponto de contração interna X_{cc} a partir:

$$X_{cc} = (1 - \rho\gamma)C + \rho\gamma X_{n+1}$$

Se $f(X_{cc}) < f(X_{n+1})$, aceitar o ponto X_{cc} e ir para o passo 6.

Caso contrário, $f(X_{cc}) \geq f(X_{n+1})$, ir para o passo 5.

5. **Redução:** Para $2 \leq i \leq n+1$, definir:

$$X_i = X_1 + \sigma(X_i - X_1)$$

6. **Fazer:** $k = k + 1$ e retornar ao passo 1 para iniciar uma nova iteração ou parar.

Os critérios de parada para o Simplex de Nelder-Mead geralmente avaliam a diferença entre o valor da função objetivo da melhor e da pior solução ou a redução no tamanho do simplex e os comparam a um determinado limite aceitável. Outro critério de parada comumente adotado é atingir um número máximo de iterações permitidas do algoritmo.

Neste trabalho, calculamos o desvio padrão dos valores da função objetivo calculadas nos vértices do simplex. Quando for menor que uma tolerância α_s , o ANMS é concluído. O parâmetro α_s pode ser considerado um fator de encolhimento do simplex.

A Figura 2.4 mostra um resumo das cinco operações geométricas realizadas no simplex, o qual é modificado em um espaço bidimensional. Nesse espaço, o triângulo formado pelos vértices (X_1 , X_2 e X_3) é o simplex. Sendo que, X_1 é o melhor vértice, X_3 é o pior vértice e C é o centroide. O triângulo vermelho é o simplex modificado após uma transformação geométrica.

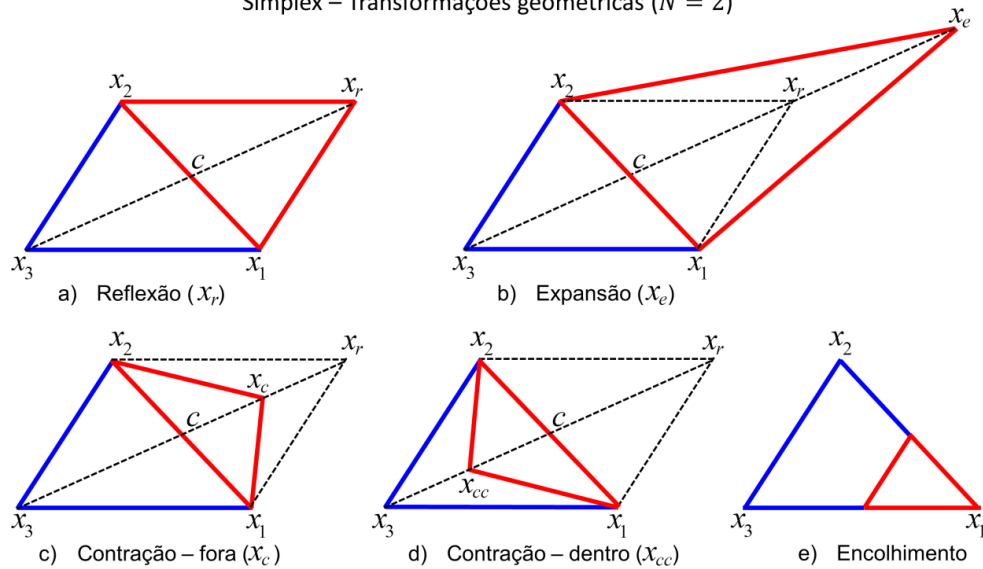
Simplex – Transformações geométricas ($N = 2$)

Figura 2.4 – Movimentos definidos no Nelder-Mead Simplex original: (a) Reflexão do ponto x_3 , dada pelo vértice x_r ; (b) Expansão, representada pelo vértice x_e ; (c) Contração externa, dada pelo vértice x_c ; (d) Contração interna, definida pelo vértice x_{cc} ; (e) Encolhimento, representado por um triângulo menor.

O Simplex de Nelder-Mead é fácil de implementar e rápido de executar. No entanto, sua versão padrão apresenta algumas desvantagens:

- Conforme mencionado anteriormente, torna-se ineficiente à medida que a dimensão do problema cresce, ou seja, escala mal com a dimensão do problema de otimização;
- Em alguns casos, o algoritmo pode convergir para pontos que não são nem o máximo nem o mínimo da função objetivo;
- Dificuldades em escolher um bom critério de parada para o Nelder-Mead Simplex e sua lenta convergência para alguns tipos de problemas foram relatadas em muitas aplicações (Givens e Hoeting, 2013);
- Além das desvantagens já conhecidas, é um método de otimização local.

Felizmente, existem muitos estudos na literatura sobre variantes do método simplex de Nelder-Mead: Lagarias *et al.* (1998) e McKinnon (1998) fizeram análises de convergência; Gao e Han (2010) e Mehta (2019) propuseram parâmetros adaptativos; e Fajfar *et al.* (2019) usou o centroide perturbado para melhorar o desempenho para problemas com dimensões maiores no método Nelder-Mead Simplex. Esses estudos contribuíram para tornar Nelder-

Mead Simplex uma técnica mais robusta, confiável e competitiva para resolver problemas de otimização não linear.

Existem várias abordagens para determinar os vértices do simplex inicial: a busca aleatória do simplex inicial em uma hiper-caixa delimitada pelas restrições do problema (limitação de borda ou contorno); o uso de simplex inicial relativamente grande; e a construção de um simplex inicial em um espaço n -dimensional a partir de um ponto \mathbf{P}_1 e n pontos, cada um marcado a uma distância de $\beta \times range_j(\Omega)$ a partir de \mathbf{P}_1 ao longo dos eixos coordenados, para obter os vértices restantes do simplex.

O coeficiente de tamanho do passo ou, simplesmente, tamanho do passo β é um fator dependente do problema. Depende dos valores de limitação para cada dimensão. Este procedimento forma um simplex retangular em \mathbf{P}_1 (simplex quase regular), preservando a qualidade do simplex.

$$\mathbf{P}_j = \mathbf{P}_1 + (0, \dots, \beta \times range_j(\Omega), \dots, 0), \quad 1 \leq j \leq n \quad (2.37)$$

No uso do ANMS como um método de busca local combinado com um algoritmo de otimização global (meta-heurística), pode-se usar um simplex inicial relativamente menor, como sendo um simplex quase regular. Tal procedimento foi adotado no presente trabalho.

2.5 Algoritmo de agrupamento K-means

O algoritmo K-means é um dos métodos particionais mais conhecidos, desenvolvido por J. A. Hartigan em 1975. Esse algoritmo particiona de forma iterativa uma base de dados com n objetos em um número predefinido K de grupos (*clusters*), utilizando a distância euclidiana de cada objeto ao centro do grupo mais próximo como critério para a alocação ou realocação dos mesmos. O objetivo do algoritmo é encontrar a melhor divisão, de forma que a soma das distâncias entre os dados de um grupo e seu respectivo centro seja minimizada para todos os grupos (Hartigan and Wong, 1979).

Matematicamente, o algoritmo K-means visa minimizar a soma dos erros quadráticos nos grupos, de acordo com a seguinte função objetivo que corresponde ao momento de inércia total, dado por:

$$I = \sum_{i=1}^k \sum_{\mathbf{X} \in Cl_i} \|\mathbf{X} - \mathbf{C}_i\|_2^2 \quad (2.38)$$

onde $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$ é o conjunto de n objetos no espaço, que representam os pontos de dados; k , número predefinido de grupos, determinando o conjunto de *clusters* $\{Cl_1, Cl_2, \dots, Cl_k\}$, \mathbf{C}_i é o centro geométrico, ou seja, é a média dos pontos pertencentes ao i -ésimo *cluster*.

O algoritmo K-means inicia selecionando aleatoriamente um conjunto de centroides no espaço n -dimensional. Cada ponto do conjunto de dados é atribuído ao cluster cujo centróide está mais próximo em termos de distância Euclidiana. Com base nessa atribuição, um novo centróide é recalculado através da média de todos os pontos que se encontram no cluster. Em seguida, uma nova atribuição é feita entre o conjunto de dados e os novos centroides mais próximos. O processo é repetido até que não haja mais alteração na atribuição dos pontos de dados, resultando na formação dos clusters finais.

De acordo com Perumal e Dilip (2011), o algoritmo de agrupamento K-means pode ser descrito em um conjunto de etapas:

1. Escolha arbitrariamente K pontos do conjunto de dados. Esses pontos representam os centroides iniciais dos K grupo.
2. Atribua cada ponto do conjunto de dados ao grupo que possui o centróide mais próximo.
3. Quando todos os pontos tiverem sido atribuídos, recalcule as posições dos K centroides.
4. Repita as etapas 2 e 3 até que não haja mais mudanças significativas nas posições dos centroides em relação às posições anteriores.

A Figura 2.5 apresenta um esquema ilustrativo do funcionamento do algoritmo K-means. Nesta figura, há uma base de dados composta por 18 objetos em um espaço bidimensional, inicialmente desordenados. O algoritmo K-means é aplicado para particionar esses dados em três *clusters* distintos, cada um representado por um centroide específico que caracteriza os pontos mais próximos a ele. A Figura 2.5(a) apresenta os dados não classificados (pontos pretos) e os centroides (cruzes coloridas) escolhidos aleatoriamente. A Figura 2.5(b) mostra a atribuição de cada ponto ao centroide mais próximo. As Figuras 2.5(c), 2.5(d) e 2.5(e) apresentam os cálculos e recálculos dos centroides, bem como as novas atribuições de cada ponto ao centroide mais próximo, conforme o algoritmo de K-means avança nas iterações. Finalmente, a Figura 2.5(f) mostra novamente a atribuição de cada ponto ao centroide mais próximo e, a partir desse momento, os centroides não se movem mais, encerrando o algoritmo K-means.

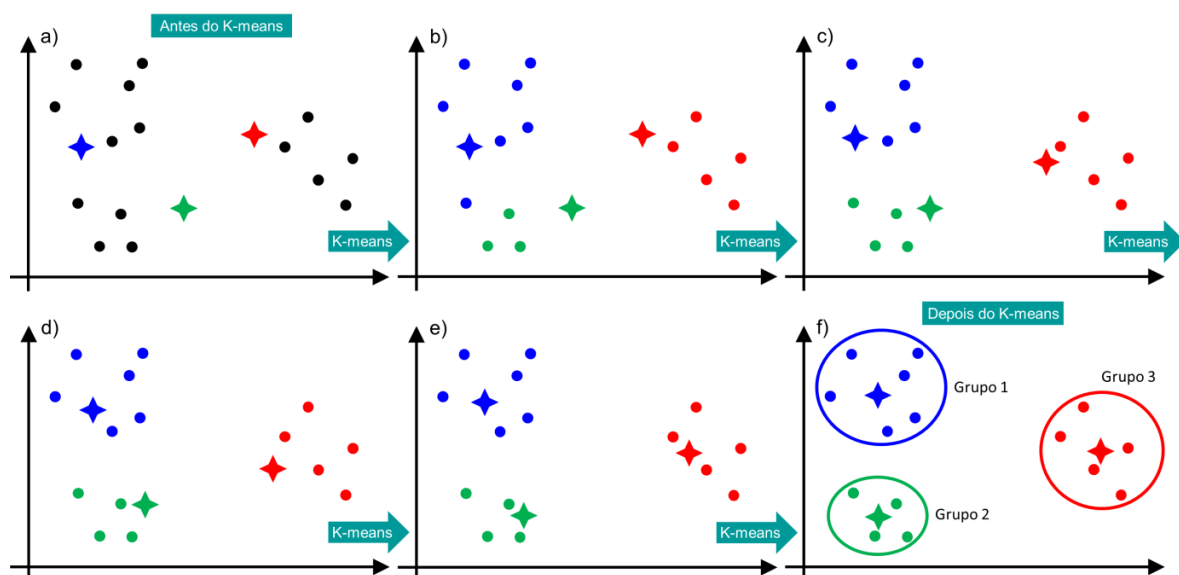


Figura 2.5 – Esquema ilustrativo do algoritmo de clusterização K-means. (a) Seleção aleatória dos centroides representados pelas cruzeiras coloridas; (b) até (e) Atribuição dos pontos de dados aos *clusters* mais próximos e atualização dos centroides correspondentes; (f) Resultados da clusterização com os grupos representados pelas cores azul, verde e vermelho."

Nas aplicações práticas do algoritmo K-means, duas questões são importantes. A primeira é o número ideal de *clusters*, K , e a segunda são os centroides iniciais. Geralmente, o número de *clusters* é definido manualmente e os centroides iniciais são escolhidos aleatoriamente no espaço n -dimensional. Durante o processo iterativo de minimização da

soma dos erros quadráticos dentro dos grupos, conforme apresentado na Equação (2.38), os centroides são realocados para garantir que cada ponto seja atribuído ao *cluster* correto.

2.6 Amostragem aprimorada do espaço de parâmetros

A construção de soluções iniciais é uma etapa crucial em muitos métodos heurísticos. Uma amostragem inadequada do espaço de parâmetros pode prejudicar severamente a convergência e a precisão da maioria dos algoritmos dessa classe. No entanto, existem várias técnicas utilizadas para mitigar esse problema. A amostragem estratificada é uma estratégia amplamente utilizada para melhorar a convergência de algoritmos baseados no método de Monte Carlo. A eficiência de uma técnica de estratificação depende principalmente da coerência dos estratos. Steigleder e McCool (2003) aplicaram a curva de Hilbert como uma técnica de estratificação para gerar amostras estratificadas em espaços n-dimensionais. Com essa técnica, é possível desenhar um número arbitrário de amostras estratificadas de forma eficiente, utilizando apenas estratificação unidimensional.

A curva de Hilbert é uma técnica que mapeia pontos com coordenadas multidimensionais para uma linha reta e vice-versa, preservando a proximidade. Essa técnica pode ser utilizada para reduzir problemas multidimensionais para unidimensionais (Lawder e King, 2001). Isso facilita a amostragem de valores no espaço paramétrico, permitindo uma distribuição que mapeia para todo o espaço de busca e, assim, promovendo maior diversidade amostral.

Neste trabalho, a curva de Hilbert foi aplicada para linearizar o espaço de parâmetros permitidos e dividi-lo em regiões com a mesma quantidade de amostras. Essa abordagem permite gerar aleatoriamente uma população inicial para o PSO com maior diversidade, de forma eficiente e fácil, contribuindo para a obtenção de soluções iniciais mais diversas e dificultando que qualquer um dos algoritmos fique preso em mínimos locais devido ao chute inicial.

O procedimento de amostragem aprimorada adotado segue as seguintes etapas:

- 1) Dado a dimensão n do problema e o grau de refinamento do preenchimento n_h , gera-se a curva de Hilbert;
- 2) Divide-se o número de partículas pelo número de regiões da curva de Hilbert;
- 3) Sorteia-se aleatoriamente essa quantidade de partículas para cada região, individualmente.
- 4) Aplica-se o operador inverso da curva de Hilbert, voltando para o espaço n -dimensional com as devidas posições das amostras nesse espaço.

Essa abordagem foi utilizada para todos os problemas, algoritmos e testes realizados neste trabalho.

2.6.1 Curva de Hilbert

A curva de Hilbert, descrita pela primeira vez em 1891 pelo matemático David Hilbert, é uma curva contínua que preenche sequencialmente todo um espaço n -dimensional, também conhecida como curva de preenchimento do espaço de Hilbert. A curva de Hilbert preserva razoavelmente bem a proximidade dos vértices (pontos). Isso significa que dois pontos de dados que estão próximos um do outro no espaço unidimensional também estão próximos um do outro após a dobra no espaço n -dimensional. Entretanto, o inverso nem sempre é verdadeiro, o que é inevitável ao passar de uma dimensão maior para uma menor. Apesar disso, a curva tende a preservar a localidade o máximo possível, o que faz dela uma ferramenta valiosa em diversas aplicações. (Cataldi *et al.*, 2021; Moon *et al.*, 2001; de Freitas *et al.*, 2019). A curva de Hilbert tem muitas aplicações e, recentemente, Araújo *et al.* (2021) a aplicaram para reordenar os elementos da malha em memória para uso com o Método do Elemento Espectral, com o objetivo de obter menos falhas de cache, melhor localidade de dados de referência e execução mais rápida. Além disso, é aplicado como base heurística para o problema do caixeiro viajante, que consiste em descobrir a rota que minimiza a jornada total.

O preenchimento do espaço ocorre em 2^n regiões distintas, que são ligadas sucessivamente (Lawder e King, 2001). O número de pontos ou vértices por região utilizados pela curva de Hilbert para preencher o espaço é dado por $2^{n(n_h-1)}$, em que $n_h \geq 2$ é a ordem da curva de Hilbert, ou seja, o grau de refinamento que determina o quão próximos os pontos estarão uns dos outros. A Figura 2.6(a) mostra um exemplo da curva de Hilbert em três dimensões (3D), formada por 8 regiões diferenciadas por cores, cada uma contendo 64 pontos. É fácil notar que $n=3$ e $n_h=3$. A Figura 2.6(b) apresenta um exemplo da curva de Hilbert em duas dimensões (2D). Neste caso tem-se $n=2$ e $n_h=5$, produzindo uma curva em 4 regiões, com cores diferentes, que contêm 256 vértices por região.

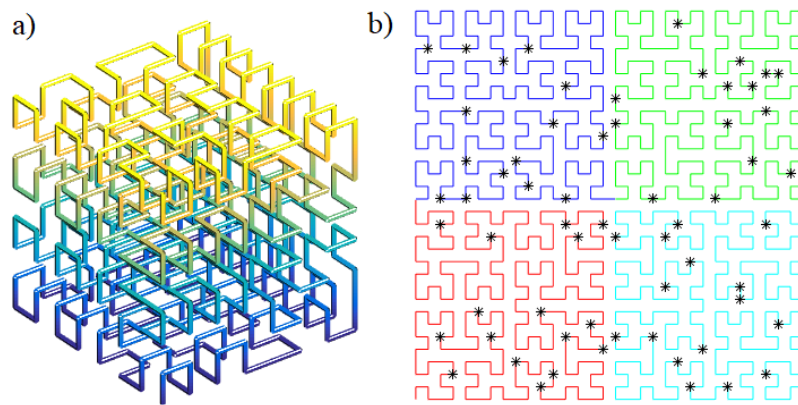


Figura 2.6 – Exemplos das curvas de Hilbert: (a) Curva de Hilbert em 3D e (b) Curva de Hilbert em 2D com as partículas distribuídas aleatoriamente nas regiões.

2.7 Literatura relacionada

Na tentativa de melhorar o desempenho da otimização não linear, novas variantes do PSO têm sido continuamente desenvolvidas e implementadas com sucesso em diversas áreas de pesquisa. Uma delas é a otimização híbrida, que combina estratégias de PSO com outros tipos de otimizadores. Por exemplo, Fan, Liang e Zahara (2004) propuseram uma abordagem híbrida usando os algoritmos Nelder-Mead Simplex e PSO para a otimização de funções multimodais. Sua principal vantagem é a facilidade de implementação numérica. Um conjunto de 17 funções de teste mostrou que a abordagem híbrida foi superior aos dois algoritmos que a compõem, separadamente, em termos de precisão da solução e taxa de convergência. Além disso, o referido método híbrido também foi comparado a outros oito métodos conhecidos,

como Algoritmo Genético (GA) híbrido, GA contínuo, Simulated Annealing (SA) e Tabu Search (TS). No geral, a nova abordagem provou ser extremamente eficaz e eficiente na busca de soluções ideais para aplicações com funções multimodais.

Rana, Jasola e Kuma (2010) propuseram uma abordagem de cluster sequencial híbrida, usando um PSO em sequência com um algoritmo K-means, para agrupamento de dados. Nessa abordagem, o PSO também foi empregado para encontrar o centro do cluster antes de um agrupamento K-means. A nova abordagem superou as desvantagens de ambos os algoritmos, melhorou o agrupamento e evitou ficar preso na solução local. Experimentos com quatro conjuntos de dados diferentes compararam K-means e PSO sozinho contra o híbrido K-means combinado com PSO e com GA. Os resultados mostraram que o algoritmo proposto gera resultados de cluster mais precisos, robustos e melhores.

Koduru, Das e Welch (2006) também apresentaram um algoritmo híbrido usando o Nelder-Mead Simplex e o PSO, mas com uma abordagem diferente, incluindo o algoritmo K-means clustering. Eles analisaram os dois algoritmos híbridos com e sem agrupamento K-means e mostraram que ambas as abordagens levaram a uma aceleração significativa na taxa de convergência para vários problemas de referência amplamente conhecidos, bem como para o problema de ajustar um modelo de genético com dados observáveis.

Firouzi, Sadeghi e Niknam (2010) propuseram um algoritmo híbrido, que combina técnicas de Otimização por Enxame de Partículas (PSO), Simulated Annealing (SA) e k-means, chamado de PSO-SA-K, que tem como objetivo superar as limitações do algoritmo k-means e auxiliar na busca por uma partição de cluster mais precisa e eficiente. A eficiência da técnica de agrupamento proposta foi avaliada com vários conjuntos de dados de referência. Os resultados da simulação mostram que o algoritmo proposto supera muitos outros métodos, como PSO, SA, combinação do PSO e SA (PSO-SA), Otimização por Colônia de Formigas (ACO), combinação do k-means e PSO (K-PSO), combinação da busca simplex de Nelder Mead e PSO (NM-PSO), K-NM-PSO, Algoritmo Genético (GA), Busca Tabu (TS), Otimização de Acasalamento de Abelhas (HBMO) e K-means para o problema de agrupamento particionado.

Nayak *et al.* (2015) propuseram uma hibridização entre os algoritmos PSO aprimorado (IPSO) e GA, juntamente com o algoritmo de agrupamento K-means, este último com o propósito de melhorar a convergência. Na primeira etapa, o IPSO foi utilizado para obter uma solução global entre os melhores centros de *clusters*. Em seguida, foram empregadas sequências de cruzamento do GA para melhorar a qualidade das partículas, enquanto a mutação foi aplicada para diversificar a busca no espaço de soluções, evitando convergência prematura. O desempenho do algoritmo híbrido proposto foi comparado com outras técnicas de agrupamento existentes, como o K-means, GA-K-means e PSO-K-means.

Perumal e Dilip (2018) propuseram uma técnica híbrida baseada na combinação do algoritmo K-means, algoritmo de busca gravitacional, busca simplex Nelder-Mead e otimização por enxame de partículas, chamada KM-GSA-NM-PSO. O método ajudou o algoritmo K-means a escapar dos mínimos locais e também aumentou a taxa de convergência do algoritmo GSA. Os autores especializaram seu algoritmo para melhorar os centros de cluster em conjuntos de dados arbitrários. O desempenho do algoritmo híbrido é comparado com os dos algoritmos K-means e KM-GSA.

Recentemente, Fakhouri, Hudaib e Sleit (2020) propuseram um algoritmo híbrido que utiliza o algoritmo PSO, o algoritmo seno-cosseno (SCA) e a técnica de otimização simplex Nelder-Mead, com o objetivo de superar as desvantagens do algoritmo PSO, como localizar mínimos locais em vez de mínimos globais, baixa taxa de convergência e baixo equilíbrio entre exploração e exploração. O desempenho do novo algoritmo em um conjunto de 23 funções unimodais e multimodais conhecidas foi superior ao obtido pelo PSO e por outros algoritmos mais avançados. O algoritmo híbrido também foi testado para resolver um problema de projeto de engenharia, como compressão de mola e viga soldada. Isso demonstrou que, em problemas de aplicação de engenharia, o algoritmo proposto tem uma boa resposta e pode ser usado em casos mais difíceis.

Outras abordagens tiveram sucesso em aumentar a taxa de convergência, melhorando a diversidade de soluções. De acordo com Nadimi-Shahraki, Taghian e Mirjalili (2021), o algoritmo de otimização de lobo cinzento aprimorado (I-GWO) introduz uma estratégia de movimento denominada estratégia de busca baseada em aprendizado de dimensão (DLH), inspirada no comportamento de caça individual dos lobos na natureza. Isso seleciona o

candidato das estratégias de busca GWO ou DLH com base na qualidade das novas soluções. A cooperação entre essas duas estratégias de busca melhora a capacidade de busca global e local do algoritmo. Semelhante a Fakhouri, Hudaib e Sleit (2020), o I-GWO foi testado em 29 funções de referência e em uma variedade de problemas de projeto de engenharia, apresentando excelentes resultados em termos de precisão e taxa de convergência. Assim, o uso de versões aprimoradas de algoritmos conhecidos deve inspirar o desenvolvimento de algoritmos híbridos mais eficazes.

Capítulo III

Metodologia

3 Metodologia

Neste capítulo, serão apresentados o algoritmo híbrido proposto e a sua avaliação aplicada individualmente em doze funções de referência. O objetivo dessa avaliação foi validar a metodologia do algoritmo híbrido, comparando os seus resultados com os obtidos pelos métodos ANMS, PSO clássico e PSO modificado. Serão descritos os resultados da avaliação em relação a apenas duas dessas doze funções de referência.

3.1 Algoritmo híbrido

O algoritmo PSO tradicional atinge maior diversidade ao explorar o espaço de busca por uma melhor solução global do que vários métodos de otimização. No entanto, ele requer muitas partículas em um processo ideal, o que aumenta seu tempo computacional. Além disso, o PSO apresenta convergência lenta quando próximo da solução ótima (Kennedy, 1997). Por outro lado, o algoritmo ANMS apresenta convergência rápida, mas cai facilmente em um ótimo local. Considerando as vantagens individuais de cada técnica, pressupõe-se que a combinação delas resulte em uma boa estratégia de otimização híbrida. Por exemplo, o PSO poderia começar como um otimizador global, e então o ANMS atuaria como um otimizador local, direcionando o processo de busca para a solução ótima, conforme já utilizado por outros autores (Fan *et al.*, 2004 e Koduru *et al.*, 2006).

Assim, nesta pesquisa, foi proposta uma nova técnica de otimização híbrida de duas fases, denominada PSO-Kmeans-ANMS. Na primeira fase, são utilizados os algoritmos PSO e K-means. O enxame de partículas do PSO é supervisionado continuamente pelo K-means, que as organiza em *clusters*. A expectativa é que, ao final dessa fase, o enxame (ou grande parte dele) seja atraído para uma região que possivelmente contenha o ótimo global (mínimo em nossa aplicação). Já na segunda fase, utiliza-se o algoritmo ANMS, que constrói um simplex a partir do melhor ponto encontrado pelo PSO na primeira fase e realiza as iterações de busca pela solução ótima. Aguarda-se que na segunda fase ocorra uma rápida convergência das partículas para o ponto ótimo da região para a qual foram atraídas na fase anterior. Nessas condições, a técnica ANMS é precisa e possui uma taxa de convergência mais rápida do que a

maioria dos algoritmos de busca direta conhecidos. O objetivo é que esse procedimento seja rápido e preciso, convergindo para uma solução ótima.

Neste estudo, o algoritmo K-means divide o enxame em 2 *clusters* a cada iteração. O salto da Fase 1 (PSO) para a Fase 2 (ANMS) ocorre através do K-means quando um dos *clusters* se torna dominante, ou seja, muito maior que o outro. Para evitar a convergência prematura do PSO modificado na Fase 1, o K-means só entra em ação após 50% do número máximo de iterações, It_{max} , terem sido realizadas. Após essa restrição, é permitido supervisionar o algoritmo de otimização por meio do K-means para uma possível mudança de fase. O tempo do K-means, representado por T_{Km} é calculado como $T_{Km} = 0,5 \times It_{max}$. Os valores 50% e It_{max} são parâmetros dependentes da aplicação.

Ainda na Fase 1 da otimização híbrida, em algum momento do processo iterativo após T_{Km} , o enxame pode atingir uma distribuição de partículas bastante homogênea formando uma “bola” que se contrairá durante as próximas iterações. Quando isso acontece, o algoritmo K-means pode não ser capaz de determinar um *cluster* dominante, fazendo com que o PSO vá para o final das iterações permitidas. Na tentativa de evitar esse problema, um mecanismo de monitoramento da homogeneidade do enxame é acionado a cada iteração. O monitoramento é baseado na razão entre os desvios padrão da aptidão das partículas do enxame no momento atual, $std(A)$, e inicial, $std(AI)$. Se essa razão for menor ou igual a um determinado valor, $rel \leq rel_{st}$, ocorrerá a mudança da Fase 1 para a Fase 2. Onde,

$$rel = \frac{\text{tamanho do maior cluster}}{\text{tamanho do menor cluster}} = \frac{std(A)}{std(AI)},$$

e rel_{st} é um parâmetro que depende da aplicação. Na presente pesquisa, foi definido que $rel_{st} = 0,25$.

Em termos gerais, o PSO-Kmeans-ANMS tenta iterativamente melhorar as soluções candidatas para um determinado problema de otimização, no que diz respeito a uma dada

função objetivo $f(\mathbf{X})$, onde cada solução \mathbf{X}_i^k é representada por uma partícula i no espaço de busca n -dimensional, \Re^n , na iteração de ordem k .

Para a Fase 1, os parâmetros adaptativos relacionados ao algoritmo PSO são: o fator de inércia (w), os coeficientes de aceleração (C_1 e C_2), a porcentagem de partículas repulsivas (rr_p), o limite permitido para a relação entre os desvios padrão dos enxames atual, rel_s , e inicial, rel_{st} e o limiar da rebelião (τ). Além disso, os parâmetros fixos n_{Pop} , It_{max} e η representam respectivamente o tamanho da população do enxame, o número de iterações permitidas e a porcentagem de fixação da velocidade. Para o algoritmo K-means, é necessário definir o número de *clusters* no enxame de partículas.

Para a Fase 2, referente ao algoritmo NM, os parâmetros ρ , χ , γ e σ correspondem aos coeficientes de reflexão, expansão, contração e redução, respectivamente. Além disso, outros parâmetros nessa fase são β e α_s , com o primeiro sendo o tamanho do passo para formar o simplex a partir de um ponto \mathbf{P}_1 ; e o segundo, a tolerância para o critério de parada. Finalmente, é necessário definir Ω , rel_s e T_{Km} , representando a hipercaixa que delimita o espaço de busca, a medida da razão de tamanho e o tempo K-means, respectivamente.

O pseudocódigo do algoritmo PSO-Kmeans-ANMS, para otimizar uma função objetivo $f(\mathbf{X}): \Re^n \rightarrow \Re$ por meio de cada iteração k é apresentado a seguir:

0. **Inicialização:** Definir $k=1$; usar a curva de Hilbert para mapear todas as posições \mathbf{X}^k e velocidades $\mathbf{V}^k=0$, inicializadas aleatoriamente, para todo o enxame; e calcular a fixação da velocidade

$$V_{j,max} = \eta \times range_j(\Omega), \quad j=1, \dots, n$$

Início da Fase 1

1. Atualizar os parâmetros w , C_1 , C_2 , rr_p e τ usando, para cada um, deles a seguinte Equação:

$$S = S_{inicial} - \left(\frac{It}{It_{max}} \right) (S_{inicial} - S_{final})$$

2. Determinar o número de partículas rebeldes, n_{reb} :

$$n_{reb} = \left(\frac{rr_p}{100} \right) \times n_{Pop}$$

3. Determinar o valor da variável dir :

$$dir = 1$$

Se $rand \leq \tau$, **então**:

$$dir = -1$$

Fim

4. Atualizar os valores da melhor posição individual e global:

Para cada partícula i do enxame **faça**

Calcular o valor de aptidão $fit(\mathbf{X}_i^k) = f(\mathbf{X}_i^k)$;

Se o valor de aptidão atual, $fit(\mathbf{X}_i^k)$, é menor que o do melhor individual $f(\mathbf{X}_{ib}^k)$, ou seja, $fit(\mathbf{X}_i^k) < f(\mathbf{X}_{ib}^k)$, **então**

Atualizar o valor do melhor individual \mathbf{X}_{kb}^k para receber o valor da partícula atual, $\Rightarrow \mathbf{X}_{ib}^k = \mathbf{X}_i^k$;

Fim

Se o valor de aptidão atual, $fit(\mathbf{X}_{ib}^k)$, é menor que o do melhor global $f(\mathbf{X}_{gb}^k)$, ou seja, $fit(\mathbf{X}_{ib}^k) < f(\mathbf{X}_{gb}^k)$ **então**

Atualizar o valor do melhor global \mathbf{X}_{gb}^k para receber o valor da partícula atual, $\Rightarrow \mathbf{X}_{gb}^k = \mathbf{X}_{ib}^k$;

Fim

5. Atualizar o enxame:

Para cada partícula do enxame, atualizar a velocidade considerando a atração entre as partículas:

Para $i = 1, 2, \dots, (n_{Pop} - n_{reb})$, **faça**

$$\mathbf{V}_{ij}^{k+1} = w\mathbf{V}_{ij}^k + C_1r_1(\mathbf{X}_{ib,j}^k - \mathbf{X}_{ij}^k) + C_2r_2(\mathbf{X}_{gb,j}^k - \mathbf{X}_{ij}^k)$$

Fim

Para as partículas rebeldes, incluindo as forças de repulsão ou atração:

Para $i = (n_{Pop} - n_{reb}) + 1, \dots, n_{Pop}$, **faça**

$$\mathbf{V}_{ij}^{k+1} = w\mathbf{V}_{ij}^k + dir \left[C_1r_1(\mathbf{X}_{ib,j}^k - \mathbf{X}_{ij}^k) + C_2r_2(\mathbf{X}_{gb,j}^k - \mathbf{X}_{ij}^k) \right], \quad j = 1, \dots, n$$

Fim

Atualizar a posição de todas as partículas do enxame usando:

$$\mathbf{X}^{k+1} = \mathbf{X}^k + \mathbf{V}^{k+1}$$

6. Determinar e selecionar dois clusters usando o algoritmo K-means

Aplicar o algoritmo K-means;

$$\text{Calcular } rel = \frac{std(A)}{std(AI)},$$

Se $k > T_{Km}$, **então**

Se $rel \geq rel_s$, **então**

\Rightarrow Vá para a etapa 8 (pare a Fase 1 e vá para a Fase 2)

Fim

Se $rel \leq rel_{st}$, **então**

⇒ Vá para a etapa 8 (pare a Fase 1 e vá para a Fase 2)

Fim

Fim

Caso contrário, continue na etapa 7

7. Faça $k = k + 1$ (o incremento da iteração na Fase 1).

Se $k \leq It_{max}$, então

⇒ Retornar ao passo 1 (iniciar uma nova iteração na Fase 1)

Fim

Caso contrário, continue na etapa 8 (interrompa a Fase 1 e inicie a Fase 2).

Fim da Fase 1

Início da Fase 2

8. Determinar os $n+1$ vértices do simplex \mathbf{P}_j fazendo:

$$\mathbf{P}_1 = \mathbf{X}_{gb}^k$$

$$\mathbf{P}_{j+1} = \mathbf{P}_1 + (0, \dots, \beta \times range_j(\Omega), \dots, 0), \quad \forall j = 1, 2, \dots, n$$

9. Definir $l = 1$ e aplique o algoritmo ANMS ao simplex determinado acima até atingir o critério de parada. Ou seja, para cada iteração l do ANMS, faça:

Calcule o desvio padrão da aptidão dos vértices do simplex,

$$std(\text{fit}(\mathbf{P}^l))$$

Se $std \geq \alpha_s$, então

⇒ Iniciar uma nova iteração no ANMS da Fase 2

$$l = l + 1$$

Fim

Caso contrário, pare a Fase 2

Fim da Fase 2

Fim do algoritmo

3.2 Validação do algoritmo proposto com as Funções Benchmark

Para validar o algoritmo híbrido de otimização PSO-Kmeans-ANMS, foi realizada uma avaliação individual aplicada a 12 funções de referência. Essa avaliação foi feita usando funções com aspectos diferentes, como funções com muitos mínimos locais (função de Ackley, função Rastrigin), em forma de bacia (função Sphere), em forma de prato (função Zakharov), em forma de vale (função Rosenbrock), com inclinação picos/quedas (função de Michalewicz, F12), em forma de funil (F22) e outras, como a função de Beale, a função de Styblinski-Tang e a função de Peaks. As funções F12 e F22 podem ser encontradas no trabalho de (Abualigah *et al.*, 2021), enquanto as demais aparecem em sites da Internet, como na página: *Virtual Library of Simulation Experiments: Test Functions and Datasets* (Surjanovic, 2013).

A seguir, para um melhor entendimento da funcionalidade do algoritmo híbrido de otimização desenvolvido nesta pesquisa, serão detalhados os resultados da validação com as funções Rosenbrock e Rastrigin. As descrições de todas as funções encontra-se no Apêndice. Na Tabela 3.6, são apresentados os resultados de todas as funções para um enxame com 36 partículas.

A função Rosenbrock, também conhecida como função vale, é uma função não convexa e unimodal com dimensão 2D. Ela é caracterizada pela existência de um único mínimo global em um extenso vale parabólico, cuja convergência para o mínimo é difícil de encontrar. Ou seja, é uma função muito complexa devido à sensibilidade ao chute inicial que, se estiver longe do mínimo, pode provocar a divergência do método. Além disso, independente da proximidade do ponto de mínimo global, mesmo pequenas diferenças na distância podem causar uma alteração considerável no valor da imagem.

A função Rastrigin com duas dimensões é uma função não linear multimodal caracterizada pela existência de um único mínimo global e vários mínimos locais muito próximos entre si, dificultando a varredura e a busca do ótimo global. Ou seja, ela possui

várias bacias de atração cujos pontos convergem para esses mínimos (atratores). Em ambas as funções citadas, a função $f(\mathbf{X})$ é igual à função objetivo do problema ($f(\mathbf{X}) = fit(\mathbf{X})$), enquanto no FWI é expressa por $fit(\bullet) = \phi(m)$ dado pela Equação (2.1).

Os parâmetros adotados para o PSO clássico na Fase 1 com $n_{Pop} = 36$, foram os seguintes:

- Número máximo de iterações, $It_{max} = 54$, ou seja, $It_{max} = 1,5 \times n_{Pop}$;
- Número máximo de iterações na Fase 1 para o K-means atuar, $T_{Km} = 27$, ou seja, $T_{Km} = 0,5 \times It_{max}$;
- Número máximo de avaliações, $It_{max} \times n_{Pop} = 1944$.

Ao final da Fase 1, foi adotado o valor de β correspondente a 10% do intervalo das variáveis dimensionais do espaço de busca do problema para a construção do simplex equilátero pelo algoritmo ANMS na Fase 2 do algoritmo proposto.

Os valores inicial e final para cada parâmetro do PSO em todas as aplicações estão na Tabela 3.1. Esses parâmetros são atualizados no intervalo por meio da Equação (2.32).

Tabela 3.1 – Intervalos de valores usados para cada parâmetro do PSO em todas as aplicações desta pesquisa.

Parâmetros	Valores	
	<i>Inicial</i>	<i>Final</i>
w	0,9	0,2
C_1	2,5	0,5
C_2	0,5	2,5
$rr_p(\%)$	80	20
τ	0,35	0,15

Na Fase 1 do algoritmo híbrido proposto, dois critérios de parada foram adotados: um baseado na relação entre os tamanhos dos *clusters* no instante atual, $rel \geq rel_s = 4$, e outro na relação entre os desvios padrão dos valores da função objetivo referentes aos enxames atual e inicial, $rel \leq rel_{st} = 0,25$. Ao final da Fase 2, foram empregados dois critérios de parada para o algoritmo ANMS: um baseado no número máximo de avaliações, 1944, e outro na tolerância, $\alpha_s = 10^{-4}$.

Para evitar uma convergência prematura na Fase 1, optou-se por verificar os critérios de parada, apenas após a metade do total de iterações permitidas. Lembrando que esses parâmetros são do tipo problemas dependentes.

3.2.1 Avaliação de desempenho com a função de Rosenbrock

A Figura 3.1 mostra os detalhes do processo híbrido de otimização na Fase 1 do algoritmo PSO-Kmeans-ANMS. A Figura 3.1(a) ilustra a aparência do gráfico da função Rosenbrock e o espaço de busca utilizado, acompanhado das curvas de nível da função. A Figura 3.1(b) evidencia o enxame inicial de partículas, pontos em azul, distribuídas pela curva de Hilbert no espaço de busca; e o enxame final de partículas, círculos em vermelho, encontradas pelo PSO na Fase 1. Além disso, também evidencia o ponto ótimo exato, círculo preto. A Figura 3.1(c) mostra o final da Fase 1 com 28 das 54 iterações disponíveis concluídas. Nessa ocasião, o K-means dividiu o enxame em dois grupos: o maior com 29 partículas, círculos azuis, e o menor com 7 partículas, círculos vermelhos. Nesse caso, o critério de parada da Fase 1 foi atendido pela relação entre os tamanhos dos *clusters*, ou seja, $rel_s = 29/7 > 4$. A Figura 3.1(d) mostra a curva de convergência para o logaritmo do valor da função objetivo da melhor partícula do enxame, a cada iteração, dada pelo PSO na Fase 1. A Figura 3.1(e) apresenta a curva para o logaritmo das médias entre os valores da função objetivo para todas as partículas do enxame, a cada iteração, dada pelo PSO na Fase 1. Observa-se que, devido à natureza rebelde de algumas partículas, o gráfico apresenta flutuações significativas para cima e para baixo. Concluindo, a Figura 3.1(f) mostra os valores da primeira coordenada da primeira partícula do enxame concedidos pelo PSO na Fase 1 em cada iteração.

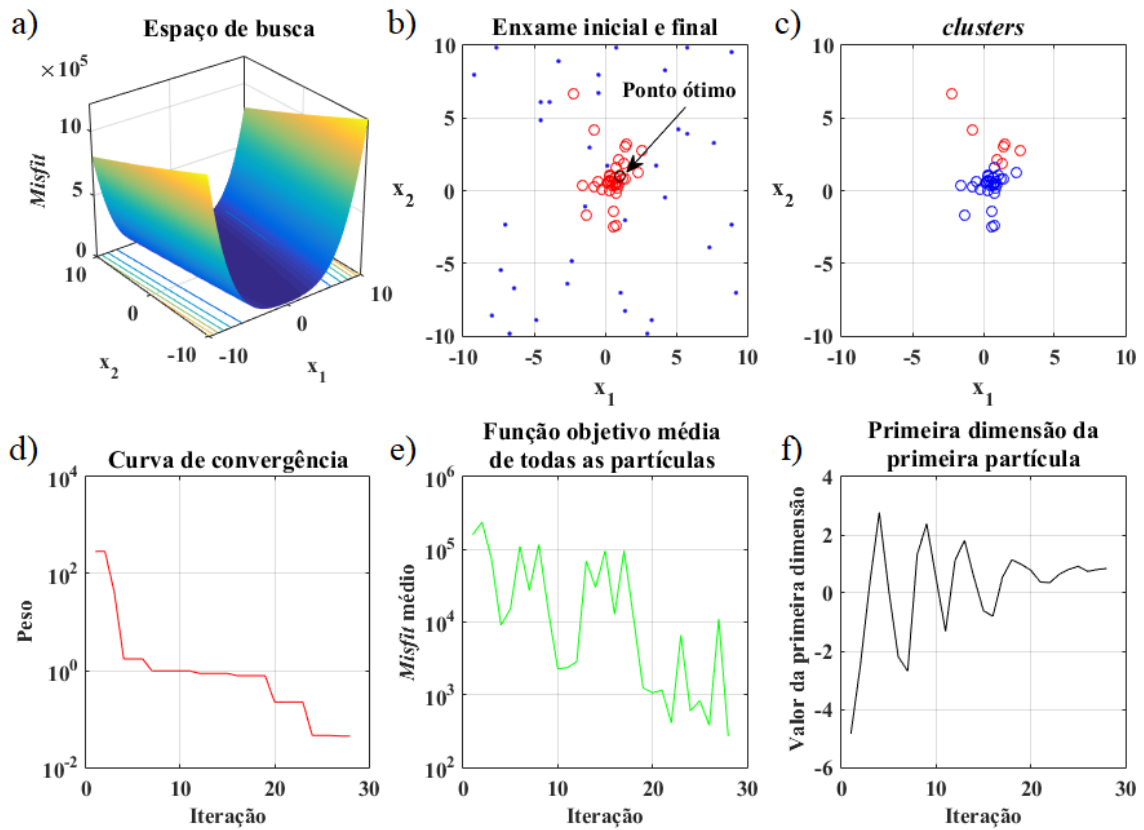


Figura 3.1 – Detalhes do processo de otimização do PSO-Kmeans-ANMS na Fase 1 para o critério de parada baseado na razão entre os tamanhos dos *clusters* no instante atual.

A Figura 3.2 apresenta os detalhes do processo de otimização na Fase 2 do algoritmo proposto. A Figura 3.2(a) mostra a curva de convergência para o logaritmo do valor da função objetivo da melhor partícula do simplex, em cada iteração, dada pelo ANMS na Fase 2. A Figura 3.2(b) exibe a curva para o logaritmo das médias entre os valores da função objetivo para todas as partículas do simplex fornecidas pelo ANMS na Fase 2 a cada iteração. A Figura 3.2(c) mostra os valores da primeira coordenada da primeira partícula correspondente ao simplex inicial, a cada iteração, dados pelo ANMS na Fase 2. Por fim, a Figura 3.2(d) apresenta o simplex equilátero inicial, círculos azuis, e o final, círculos vermelhos, encontrados pelo ANMS na Fase 2. Também mostra o ponto ótimo exato, representado por um círculo preto. Na Fase 2, o algoritmo ANMS foi executado com 64 das 1.944 avaliações das funções objetivo.

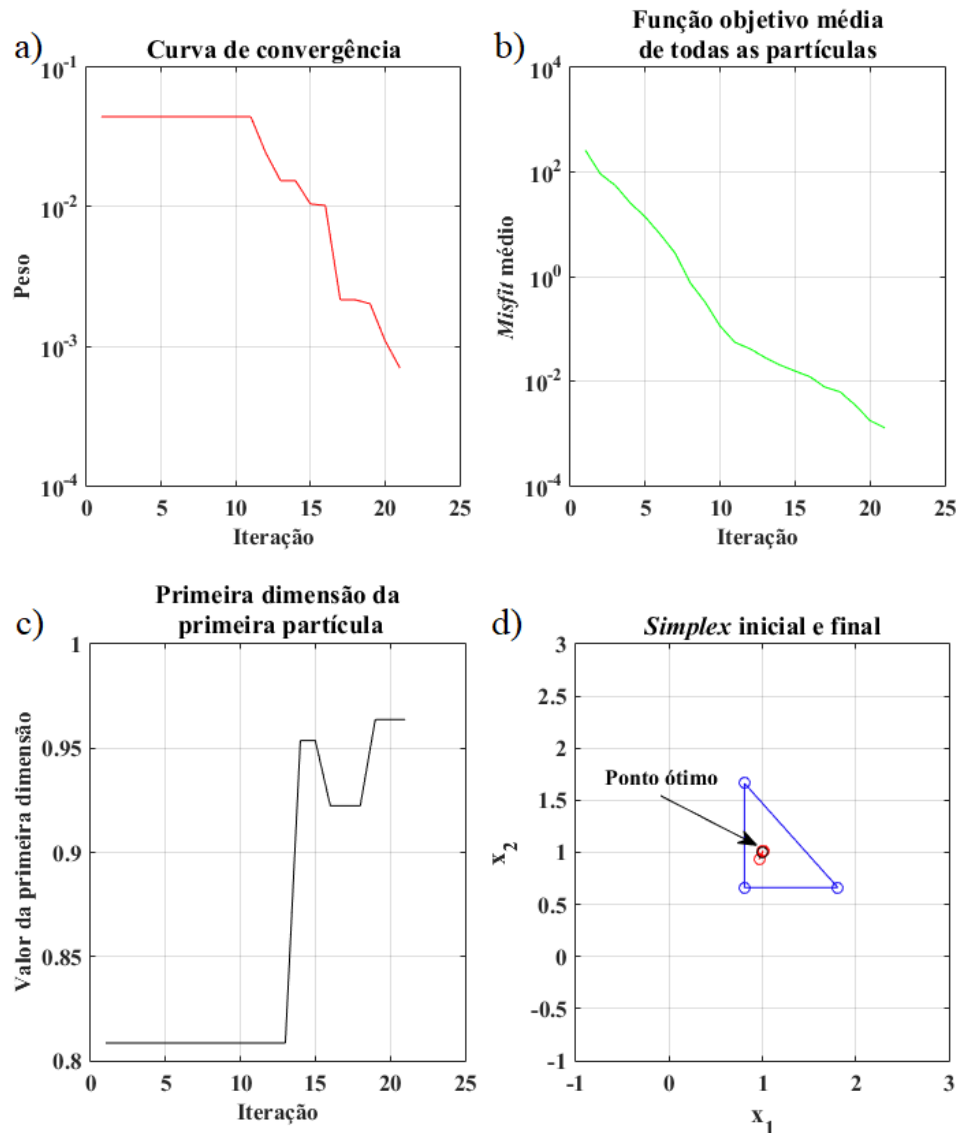


Figura 3.2 – Detalhes do processo de otimização do PSO-Kmeans-ANMS na Fase 2.

A Tabela 3.2 apresenta os detalhes dos resultados obtidos pelo algoritmo híbrido PSO-Kmeans-ANMS na otimização da função de Rosenbrock. A redução do número de avaliações da função objetivo e a precisão do ponto ótimo são as partes mais importantes dos resultados. Observa-se na tabela um total de 1048 avaliações para o algoritmo PSO-Kmeans-ANMS, sendo 1008 na Fase 1 e 40 na Fase 2, e uma boa precisão na solução encontrada. Comparando com as 1944 avaliações disponíveis para o PSO clássico, tem-se uma redução de 46,09% do número total de avaliações. Para funções objetivo que possuem alto custo computacional, como no caso do FWI, esse ganho de eficiência é bastante significativa.

Tabela 3.2 – Resultados do algoritmo PSO-Kmeans-ANMS para a função Rosenbrock.

Parâmetros	Valor	
	Obtido	Exato
x_1	1,0063	1,0
x_2	1,0153	1,0
$f(x_1, x_2)$	$6,96 \times 10^{-3}$	0,0

A Figura 3.3 mostra que na Fase 1 do algoritmo híbrido, quando o enxame forma um aglomerado de partículas muito homogêneo, o critério de parada baseado na razão entre os tamanhos dos aglomerados pode falhar. É possível observar na Figura 3.3(c) que os dois *clusters* são semelhantes e parecem ser partes de um único *cluster*. Nesse caso, torna-se difícil decidir sobre um *cluster* dominante. Nesta simulação, o K-means dividiu o enxame em dois *clusters*, um com 22 e outro com 14 partículas que estão representadas, respectivamente, pelos círculos em azul e em vermelho na Figura 3.3(c). Ou seja, o critério de parada baseado na relação entre os tamanhos dos aglomerados não foi atendido, $rel_s = 22/14 < 4$. No entanto, o critério de parada foi atendido pela relação entre os desvios padrão dos valores da função objetivo referentes aos enxames atual e inicial, $rel_{st} = 1,37 \times 10^2 \div 2,90 \times 10^5 < 0,25$.

O critério de parada baseado na razão entre os tamanhos dos *clusters* é mais interessante para funções com mínimos locais, como a função Rastrigin, enquanto que o critério baseado na relação entre os desvios padrão é interessante para funções mais suaves que possuem apenas um mínimo local, como a função de Rosenbrock.

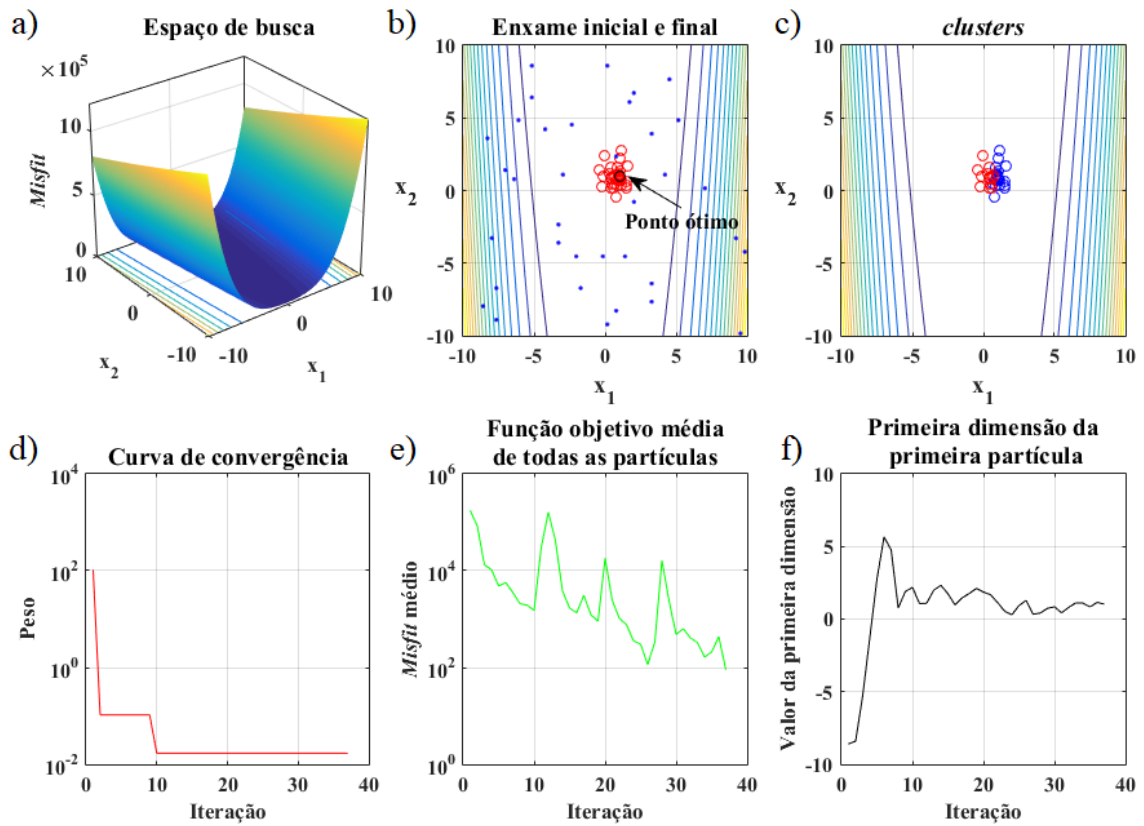


Figura 3.3 – Detalhes do processo de otimização do PSO-Kmeans-ANMS na Fase 1 para o critério de parada baseado na razão entre os desvios padrão dos valores da função objetivo referentes aos enxames atual e inicial.

3.2.2 Avaliação de desempenho com a função de Rastrigin

Na simulação de otimização da função Rastrigin pelo algoritmo PSO-Kmeans-ANMS, foram utilizados os mesmos parâmetros. No entanto, foi adotado o valor de β correspondente a 5%. Isso porque o intervalo das variáveis dimensionais foi reduzido pela metade. Como a função é multimodal, o critério de parada na Fase 1 do algoritmo é, preferencialmente, baseado na razão entre os tamanhos dos *clusters*.

As Figuras 3.4 e 3.5 mostram, respectivamente, os detalhes do processo híbrido de otimização na Fase 1 e 2 do algoritmo PSO-Kmeans-ANMS, semelhantes aos apresentados para a função de Rosenbrock.

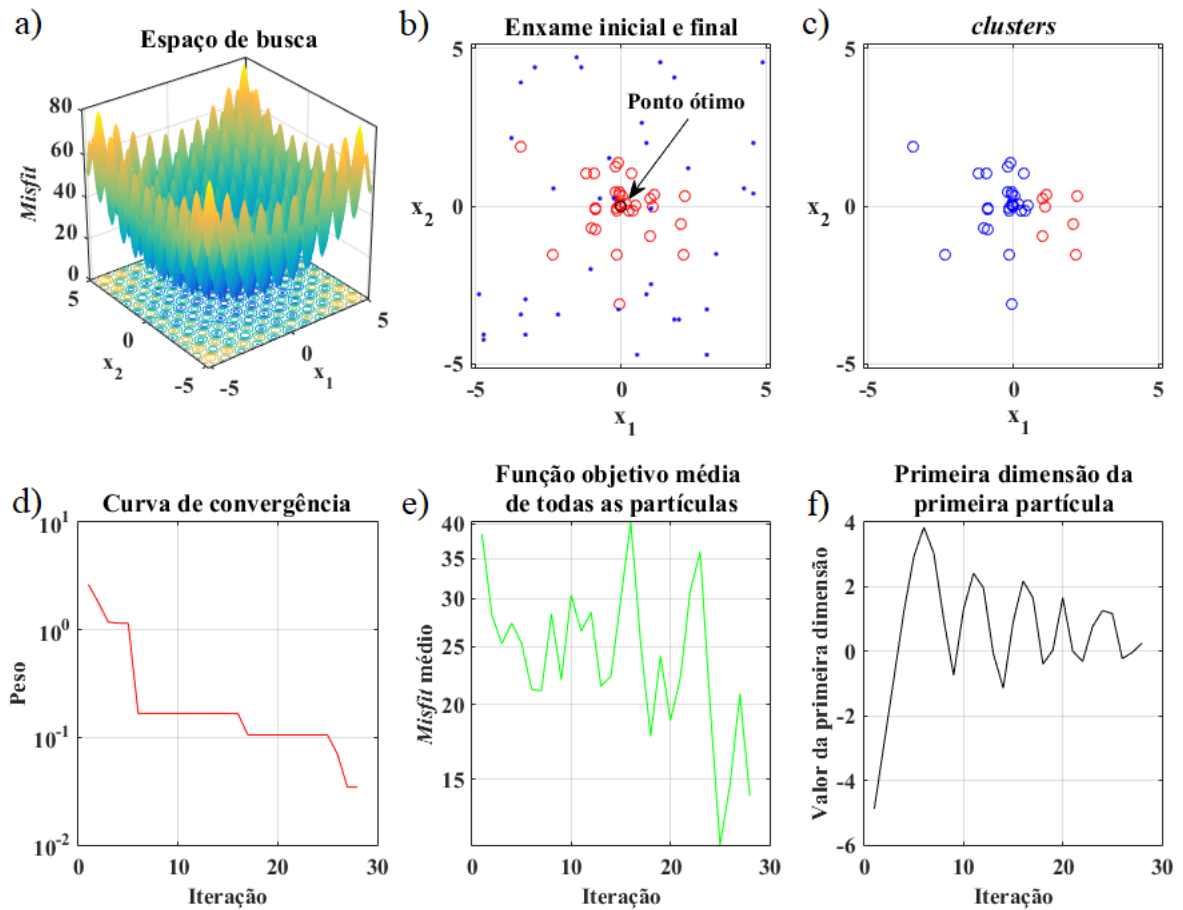


Figura 3.4 – Detalhes do processo de otimização do PSO-Kmeans-ANMS na Fase 1 para o critério de parada baseado na razão entre os tamanhos dos *clusters* no instante atual.

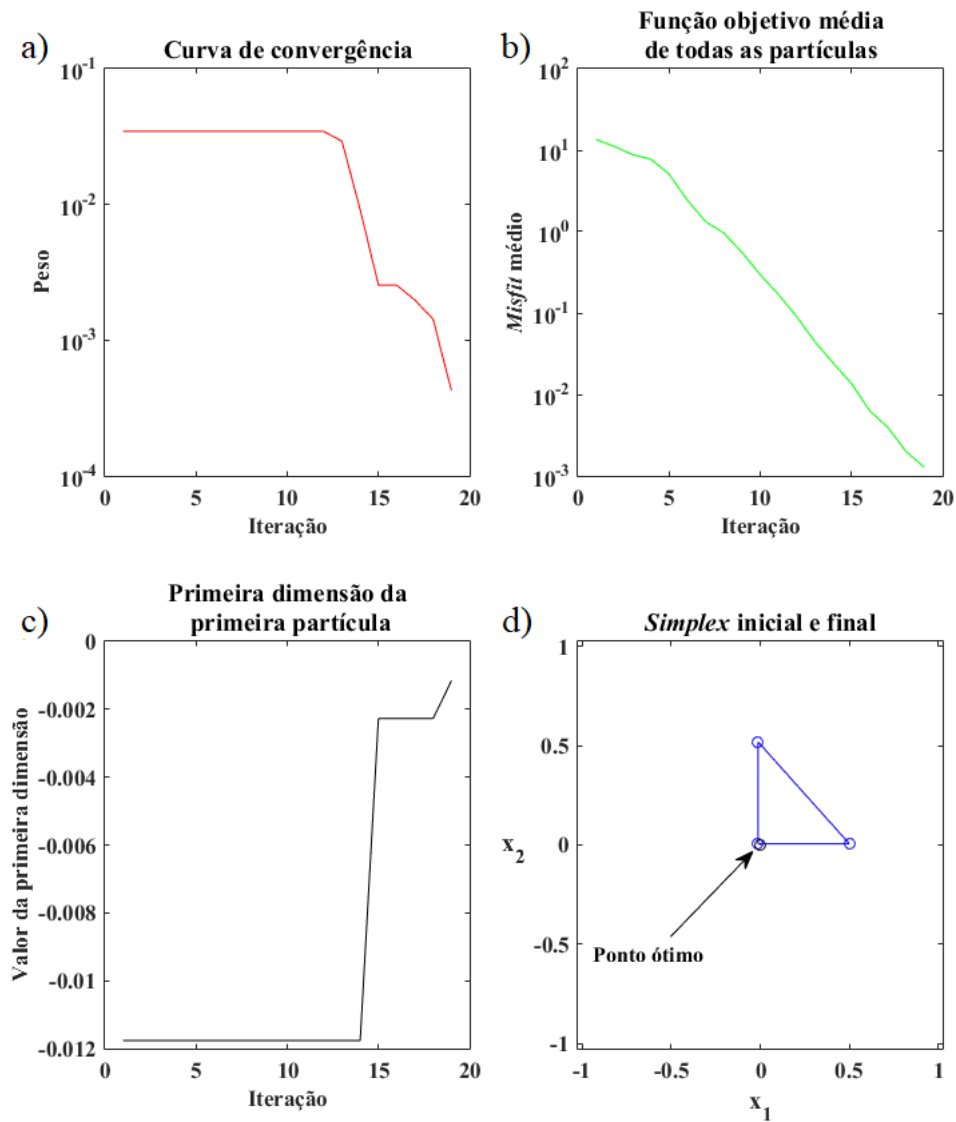


Figura 3.5 – Detalhes do processo de otimização do PSO-Kmeans-ANMS na Fase 2.

A Tabela 3.3 apresenta os detalhes dos resultados encontrados pela simulação na otimização da função Rastrigin pelo algoritmo híbrido PSO-Kmeans-ANMS. Observa-se nessa tabela um total de 1044 avaliações para o algoritmo PSO-Kmeans-ANMS, sendo 1008 na Fase 1 e 36 na Fase 2, com uma boa precisão na solução encontrada. Comparando com as 1944 avaliações para o PSO clássico, houve uma redução de 46,30% do número total de avaliações. Portanto, o resultado foi semelhante ao obtido com a função de Rosenbrock no processo de otimização.

Tabela 3.3 – Resultados do algoritmo PSO-Kmeans-ANMS para a função Rastrigin.

Parâmetros	Valor	
	Obtido	Exato
x_1	$-1,1609 \times 10^{-3}$	0,0
x_2	$8,9147 \times 10^{-4}$	0,0
$f(x_1, x_2)$	$4,25 \times 10^{-4}$	0,0

3.2.3 Análise da influência do tamanho do enxame nos resultados da otimização

A seguir, será realizada uma análise da influência do tamanho do enxame, n_{pop} , no desempenho dos algoritmos ANMS, PSO clássico, PSO modificado e PSO-Kmeans-ANMS estudados nesta pesquisa para as duas funções de referência escolhidas: Rosenbrock, que é monomodal, e Rastrigin, que é multimodal. Essa análise foi realizada com 100 simulações para cada algoritmo.

foi definida?

As Tabelas 3.4 e 3.5 apresentam a taxa de sucesso, $sr(\%)$, para cada algoritmo referente aos tamanhos do enxame com 8, 12, 20, 28 e 36 partículas. Os resultados mostram que todos os algoritmos obtiveram melhor resposta com o aumento do tamanho do enxame, exceto o ANMS, que apresentou flutuações adversas em seus resultados em relação à função multimodal Rastrigin. Além disso, o algoritmo PSO-Kmeans-ANMS foi o mais bem-sucedido em todas as situações aqui consideradas. Os resultados também mostram que a abordagem híbrida se torna mais competitiva à medida que o enxame diminui de tamanho. Esse é um bom resultado quando se trata de problemas com função objetivo de alto custo computacional, como no caso do FWI.

Tabela 3.4 – População *versus* taxa de sucesso para a função Rosenbrock.

n_{pop}	Algoritmos			
	ANMS	PSO clássico	PSO modificado	PSO-Kmeans-ANMS
8	98	32	28	87
12	99	40	43	99
20	98	84	86	100
28	100	99	96	100
36	100	100	100	100

Tabela 3.5 – População *versus* taxa de sucesso para a função Rastrigin.

n_{pop}	Algoritmos			
	ANMS	PSO clássico	PSO modificado	PSO-Kmeans-ANMS
8	4	17	25	77
12	40	40	31	99
20	60	79	58	100
28	9	96	92	100
36	9	98	99	100

A Figura 3.6 mostra a visualização gráfica dos resultados apresentados nas Tabelas 3.4 e 3.5 para todos os algoritmos avaliados com as funções Rosenbrock e Rastrigin. É possível observar que o algoritmo híbrido apresenta uma linha de tendência para as duas funções, alcançando 100% de taxa de sucesso com apenas 20 partículas. É importante destacar que o ANMS apresentou um desempenho superior para a função Rosenbrock devido, em parte, à sua característica de ser convexa e suave, o que facilita a resolução pelo algoritmo. No entanto, quando aplicado à função Rastrigin, o ANMS teve um desempenho inferior. Isso se deve à complexidade da superfície de resposta da função Rastrigin, que apresenta muitos mínimos locais e torna mais desafiador para o ANMS encontrar a solução global. Além disso, como o ANMS utiliza simplex, o tamanho da população não afeta sua taxa de sucesso. Por outro lado, tanto o PSO clássico quanto o modificado apresentaram uma tendência de sucesso crescente com o número de partículas utilizadas. No entanto, o algoritmo híbrido apresenta uma taxa de convergência significativamente melhor e é menos dependente do número de partículas utilizadas.

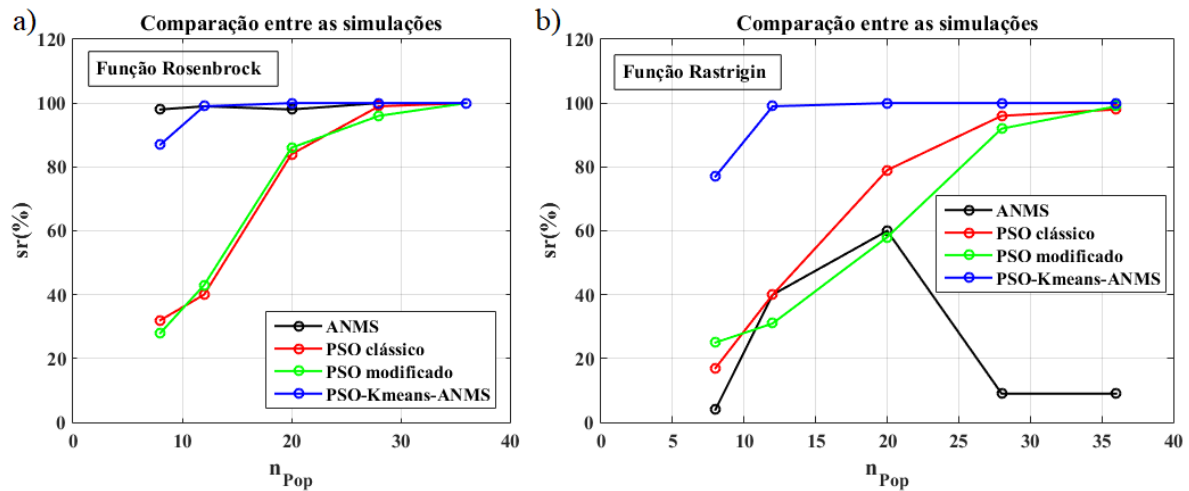


Figura 3.6 – Comparação entre as simulações para todos os algoritmos.

As Figuras 3.7 e 3.8 apresentam a dispersão espacial dos resultados das simulações realizadas para as funções Rosenbrock e Rastrigin, respectivamente. Em cada uma dessas figuras, são exibidas 100 soluções, representadas por círculos vermelhos, para cada um dos algoritmos utilizados: ANMS (a, e), PSO clássico (b, f), PSO modificado (c, g) e PSO-Kmeans-ANMS (d, h). Além disso, as simulações foram conduzidas com enxames de 8 (a, b, c, d) e 36 partículas (e, f, g, h).

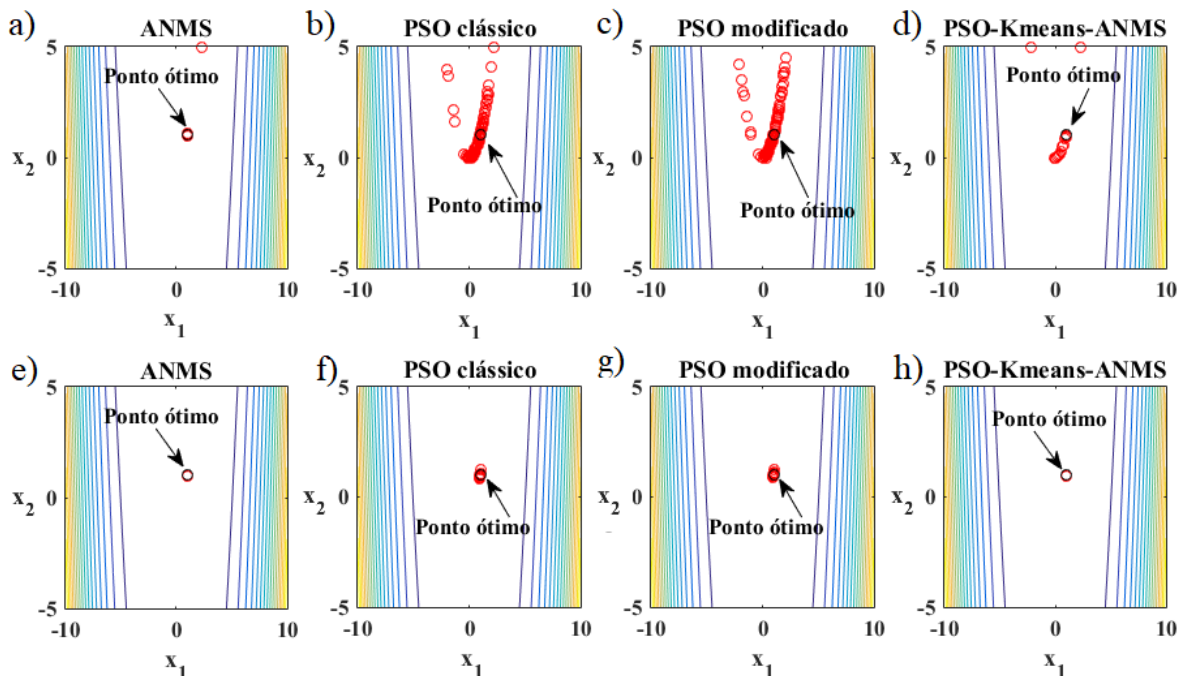


Figura 3.7 – Detalhes das simulações para a função Rosenbrock, apresentando os resultados obtidos por todos os algoritmos para os enxames com 8 (superior) e 36 (inferior) partículas, respectivamente.

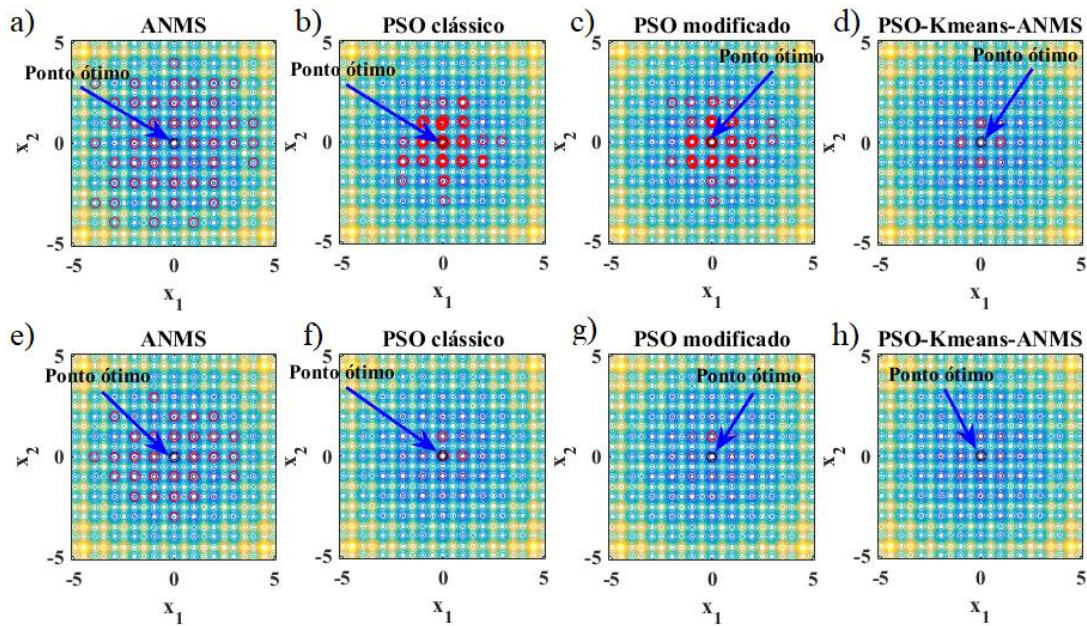


Figura 3.8 – Detalhes das simulações para a função Rastrigin, apresentando os resultados obtidos por todos os algoritmos para os enxames com 8 (superior) e 36 (inferior) partículas, respectivamente.

Tabela 3.6 – Resultados das melhores soluções para todos os algoritmos. Abreviações: $sr(\%)$, taxa de sucesso; $rt(s)$, tempo médio de execução; aes , número médio de avaliações; rop , resultado aproximado para o ponto ótimo.

Funções	Valores	Algoritmos			
		ANMS	PSO clássico	PSO modificado	PSO-Kmeans-ANMS
Ackley	$sr(\%)$	100	100	100	100
	$rt(s)$	$7,46 \times 10^{-3}$	$1,58 \times 10^{-1}$	$1,60 \times 10^{-1}$	$2,30 \times 10^{-1}$ $5,13 \times 10^{-3}$
	aes	73	1944	1944	1036 56
	rop $(x_1, x_2, f(x))$	$1,79 \times 10^{-5}$ $-1,84 \times 10^{-5}$ $7,26 \times 10^{-5}$	$-1,18 \times 10^{-4}$ $2,56 \times 10^{-5}$ $3,42 \times 10^{-4}$	$-1,90 \times 10^{-7}$ $-3,69 \times 10^{-7}$ $1,17 \times 10^{-6}$	$-1,88 \times 10^{-4}$ $-3,03 \times 10^{-5}$ $5,39 \times 10^{-4}$
Rastrigin	$sr(\%)$	26	97	98	100
	$rt(s)$	$6,87 \times 10^{-3}$	$1,54 \times 10^{-1}$	$1,55 \times 10^{-1}$	$2,57 \times 10^{-1}$ $4,18 \times 10^{-3}$
	aes	54	1944	1944	1099 41
	rop $(x_1, x_2, f(x))$	$1,47 \times 10^{-4}$ $-1,44 \times 10^{-5}$ $4,35 \times 10^{-6}$	$-2,46 \times 10^{-5}$ $-3,54 \times 10^{-6}$ $1,22 \times 10^{-7}$	$2,87 \times 10^{-9}$ $-1,84 \times 10^{-8}$ $7,11 \times 10^{-14}$	$-4,32 \times 10^{-4}$ $2,68 \times 10^{-5}$ $3,71 \times 10^{-5}$

Funções	Valores	Algoritmos			
		ANMS	PSO clássico	PSO modificado	PSO-Kmeans-ANMS
Beale	$sr(\%)$	82	86	99	100
	$rt(s)$	$4,92 \times 10^{-3}$	$1,50 \times 10^{-1}$	$1,49 \times 10^{-1}$	$2,20 \times 10^{-1}$ $2,61 \times 10^{-3}$
	aes	48	1944	1944	1011 26
	rop $(x_1, x_2, f(x))$	2,99 $4,99 \times 10^{-1}$ $5,87 \times 10^{-6}$	3,00 $5,00 \times 10^{-1}$ $1,49 \times 10^{-10}$	3,00 $5,00 \times 10^{-1}$ $8,11 \times 10^{-13}$	3,00 $5,01 \times 10^{-1}$ $2,33 \times 10^{-6}$
Rosenbrock	$sr(\%)$	100	99	100	100
	$rt(s)$	$9,21 \times 10^{-3}$	$1,46 \times 10^{-1}$	$1,47 \times 10^{-1}$	$2,22 \times 10^{-1}$ $4,03 \times 10^{-3}$
	aes	116	1944	1944	1014 45
	rop $(x_1, x_2, f(x))$	1,00 1,00 $8,05 \times 10^{-7}$	1,00 1,00 $6,69 \times 10^{-6}$	1,00 1,00 $1,72 \times 10^{-8}$	$9,99 \times 10^{-1}$ $9,98 \times 10^{-1}$ $1,00 \times 10^{-6}$
Sphere	$sr(\%)$	100	100	100	100
	$rt(s)$	$4,28 \times 10^{-3}$	$1,53 \times 10^{-1}$	$1,56 \times 10^{-1}$	$2,24 \times 10^{-1}$ $2,61 \times 10^{-3}$
	aes	33	1944	1944	1013 25
	rop $(x_1, x_2, f(x))$	$-5,55 \times 10^{-4}$ $-1,33 \times 10^{-3}$ $2,08 \times 10^{-6}$	$2,62 \times 10^{-6}$ $-4,01 \times 10^{-6}$ $2,29 \times 10^{-11}$	$-1,81 \times 10^{-7}$ $-2,77 \times 10^{-8}$ $3,35 \times 10^{-14}$	$-6,04 \times 10^{-4}$ $3,29 \times 10^{-4}$ $4,73 \times 10^{-7}$
Zakharov	$sr(\%)$	100	100	100	100
	$rt(s)$	$4,59 \times 10^{-3}$	$1,51 \times 10^{-1}$	$1,58 \times 10^{-1}$	$2,26 \times 10^{-1}$ $3,15 \times 10^{-3}$
	aes	36	1944	1944	1018 29
	rop $(x_1, x_2, f(x))$	$1,72 \times 10^{-3}$ $-3,51 \times 10^{-3}$ $6,16 \times 10^{-6}$	$-2,90 \times 10^{-5}$ $5,78 \times 10^{-5}$ $1,67 \times 10^{-9}$	$-2,13 \times 10^{-7}$ $1,32 \times 10^{-8}$ $8,52 \times 10^{-14}$	$-6,72 \times 10^{-5}$ $-1,28 \times 10^{-4}$ $4,26 \times 10^{-8}$

Funções	Valores	Algoritmos			
		ANMS	PSO clássico	PSO modificado	PSO-Kmeans-ANMS
Michalewicz	$sr(\%)$	83	100	100	100
	$rt(s)$	$4,13 \times 10^{-3}$	$1,49 \times 10^{-1}$	$1,48 \times 10^{-1}$	$2,28 \times 10^{-1}$ $2,74 \times 10^{-3}$
	aes	36	1944	1944	1056 28
	rop $(x_1, x_2, f(x))$	2,20 1,57 -1,80	2,20 1,57 -1,80	2,20 1,57 -1,80	2,20 1,57 -1,80
Styblinski	$sr(\%)$	72	100	100	100
	$rt(s)$	$4,46 \times 10^{-3}$	$1,49 \times 10^{-1}$	$1,47 \times 10^{-1}$	$2,20 \times 10^{-1}$ $3,13 \times 10^{-3}$
	aes	40	1944	1944	1030 34
	rop $(x_1, x_2, f(x))$	-2,90 -2,90 $-7,83 \times 10$	-2,90 -2,90 $-7,83 \times 10$	-2,90 -2,90 $-7,83 \times 10$	-2,90 -2,90 $-7,83 \times 10$
F12	$sr(\%)$	84	100	100	100
	$rt(s)$	$6,19 \times 10^{-3}$	$1,80 \times 10^{-1}$	$1,85 \times 10^{-1}$	$2,51 \times 10^{-1}$ $4,11 \times 10^{-3}$
	aes	39	1944	1944	1026 29
	rop $(x_1, x_2, f(x))$	$-9,99 \times 10^{-1}$ -1,00 $2,93 \times 10^{-6}$	$-10,00 \times 10^{-1}$ -1,00 $6,13 \times 10^{-10}$	-1,00 -1,00 $1,55 \times 10^{-14}$	$-10,00 \times 10^{-1}$ $-9,97 \times 10^{-1}$ $8,27 \times 10^{-7}$
F22	$sr(\%)$	100	100	100	100
	$rt(s)$	$5,43 \times 10^{-3}$	$2,00 \times 10^{-1}$	$2,00 \times 10^{-1}$	$2,55 \times 10^{-1}$ $3,51 \times 10^{-3}$
	aes	27	1944	1944	1043 21
	rop $(x_1, x_2, f(x))$	1,02 1,02 $-5,24 \times 10^{-1}$	1,02 1,02 $-5,24 \times 10^{-1}$	1,02 1,02 $-5,24 \times 10^{-1}$	1,02 1,02 $-5,24 \times 10^{-1}$

Funções	Valores	Algoritmos			
		ANMS	PSO clássico	PSO modificado	PSO-Kmeans-ANMS
F2	$sr(\%)$	100	100	100	100
	$rt(s)$	$7,62 \times 10^{-3}$	$1,57 \times 10^{-1}$	$1,63 \times 10^{-1}$	$2,33 \times 10^{-1}$ $4,68 \times 10^{-3}$
	aes	61	1944	1944	1016 44
	rop $(x_1, x_2, f(x))$	$2,31 \times 10^{-5}$ $-2,01 \times 10^{-5}$ $4,32 \times 10^{-5}$	$-3,46 \times 10^{-6}$ $3,09 \times 10^{-5}$ $3,43 \times 10^{-5}$	$1,60 \times 10^{-7}$ $5,85 \times 10^{-8}$ $2,19 \times 10^{-7}$	$-3,70 \times 10^{-5}$ $-1,41 \times 10^{-4}$ $1,78 \times 10^{-4}$
Peaks	$sr(\%)$	80	100	100	100
	$rt(s)$	$4,36 \times 10^{-3}$	$1,48 \times 10^{-1}$	$1,48 \times 10^{-1}$	$2,22 \times 10^{-1}$ $3,08 \times 10^{-3}$
	aes	38	1944	1944	1032 33
	rop $(x_1, x_2, f(x))$	$2,29 \times 10^{-1}$ -1,63 -6,55	$2,28 \times 10^{-1}$ -1,63 -6,55	$2,28 \times 10^{-1}$ -1,63 -6,55	$2,28 \times 10^{-1}$ -1,63 -6,55

3.3 Aplicação do algoritmo proposto a FWI 1D

Para demonstrar a eficiência e a precisão do algoritmo híbrido PSO-Kmeans-ANMS, foi considerado o problema não linear de Full Waveform Inversion (FWI) para um modelo sintético simples em uma dimensão, com um perfil composto por apenas duas camadas. Embora possa parecer um problema trivial, no contexto da inversão de um conjunto misto de parâmetros físicos, posição da interface (refletor) e velocidade das camadas, é um desafio difícil de ser resolvido com alta precisão. Soluções obtidas por algoritmos de otimização baseados em derivadas podem gerar um alto custo computacional. Uma alternativa é adotar uma metodologia de otimização baseada em DFO.

3.3.1 Parametrização do Modelo

Na aproximação acústica da onda sísmica, em que a densidade é considerada constante, a propriedade cinemática do meio a ser descoberto no processo de inversão é a velocidade de propagação da onda acústica c , conforme apresentado na Equação (2.11).

A Figura 3.9 mostra o modelo de velocidade sísmica 1D, no qual a linha preta sólida é o modelo verdadeiro, e as linhas tracejadas cinzas são os limites do espaço de busca. O modelo verdadeiro é composto por duas camadas com as respectivas velocidades $V_1 = 2,0$ e $V_2 = 4,0$ em unidades adimensionais por segundo (ua/s). A interface (refletor) entre essas duas camadas está localizada em $h_{rf} = 0,5$ ua. Dado que cada camada possui uma velocidade constante e a posição do refletor está bem definida, é possível parametrizar todo o modelo utilizando apenas esses três valores.

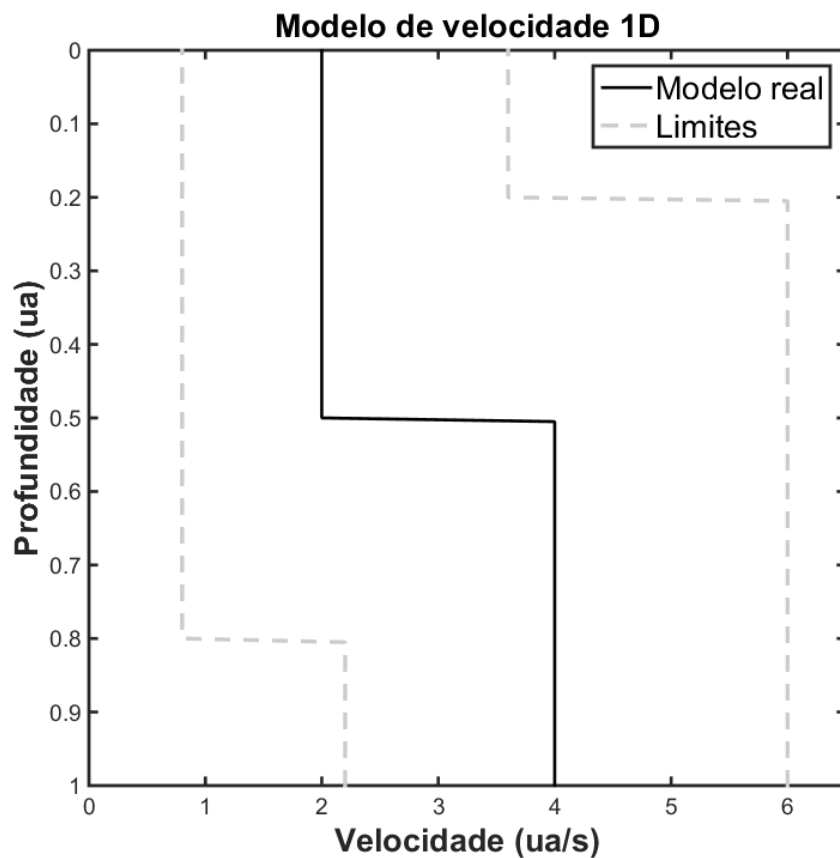


Figura 3.9 – Modelo de velocidade sísmica 1D.

Devido à maldição da dimensionalidade na otimização global, é necessário utilizar estratégias de parametrização que reduzam o número de parâmetros nos modelos estudados (Chen *et al.*, 2015). Uma parametrização é capaz de reduzir significativamente o número total de parâmetros do modelo, diminuindo assim a dimensão do espaço de busca, o que contribui para encontrar uma solução para o problema com menor custo computacional e maior precisão. Conforme investigado por Aguiar *et al.* (2016) e Gomes *et al.* (2017), uma boa parametrização pode ser obtida focando em camadas. Isso ajuda a encontrar a solução para o problema de maneira eficiente e precisa. A parametrização em função da velocidade das camadas e da posição da interface entre elas é intuitiva e fornece uma representação precisa de um determinado modelo de camadas.

Com base na definição da Equação (2.9), foi construído um operador de parametrização Γ que converte os parâmetros do modelo $\mathbf{m} = (m_1 = V_1, m_2 = V_2, m_3 = h_{rf})$ em um modelo de velocidade. Em outras palavras, esse operador cria um vetor de tamanho $n \times 1$, semelhante ao mostrado na Figura 2.2, a partir dos parâmetros que constituem o modelo \mathbf{m} , vetor esse que é exigido pela modelagem direta e que corresponde à distribuição dos parâmetros. Caso os parâmetros correspondam ao modelo real, valores exatos na Tabela 3.7, esse operador leva à construção do mesmo modelo, como mostrado na Figura 3.9.

Em muitos problemas de modelagem, é comum que os valores possíveis para um modelo sejam limitados, com base na natureza do problema. Esses limites são representados por parâmetros que devem ser definidos com base no conhecimento prévio do sistema ou em resultados de estudos anteriores. Para este trabalho específico, as restrições para os parâmetros (V_1, V_2, h_{rf}) foram definidas com base na Tabela 3.7, que apresenta os limites inferior (L_{inf}) e superior (L_{sup}) impostos para cada um deles. Durante o processo de inversão, que consiste em determinar os valores dos parâmetros que melhor representam o sistema, foram testados diversos modelos, com as linhas tracejadas em cinza na Figura 3.9 representando os limites impostos aos parâmetros no espaço de busca. Essas restrições são importantes para garantir que os valores dos parâmetros sejam realistas e coerentes com o problema em estudo, permitindo a obtenção de modelos mais precisos e confiáveis.

Na Figura 3.9, as linhas tracejadas em cinza representam os possíveis modelos testados durante o processo de inversão, ou seja, os limites impostos aos parâmetros no espaço de busca.

Quanto maior for o intervalo imposto pelos limites, maior será o esforço computacional para encontrar a solução ótima. Entretanto, se intervalo imposto pelos limites for muito pequeno, existirá o risco de deixar a solução ótima desejada fora do espaço de busca. Em termos de otimização, o modelo **m** pode ser considerado um vetor de variáveis de decisão.

Tabela 3.7 – Valores exatos dos parâmetros do modelo verdadeiro com seus respectivos limites inferior e superior.

Valores	Parâmetros		
	$V_1(du/s)$	$V_2(du/s)$	$h_{rf}(du)$
Exato	2,0	4,0	0,5
L_{inf}	0,8	2,2	0,2
L_{sup}	3,6	6,0	0,8

Apesar da imposição de restrições para os valores possíveis no espaço de parâmetros, pode-se permitir que as soluções inviáveis sejam geradas. Nesse caso, as soluções inviáveis (fora dos limites) precisam ser trazidas para as bordas da hipercaixa definida pelas restrições. Isso tende a melhorar a eficiência da busca, reduzindo o custo computacional (tempo de CPU).

3.3.2 Dados sísmicos sintéticos e modelagem

O experimento sísmico sintético foi projetado para obter dados de reflexão com um tempo de gravação máximo de 15 segundos ($t_{max} = 15s$). Para isso, foi utilizada uma fonte sísmica localizada próxima à superfície do modelo em uma posição $x_s = 0,10du$, e um receptor localizado na posição $x_r = 0,15du$. A assinatura da fonte sísmica foi dada por uma wavelet de Ricker (Ricker, 1940) com frequência de pico igual a 10 Hz. Para a modelagem direta, foi realizado um processo de discretização usando o Método das Diferenças Finitas

(MDF). No domínio espacial, a malha do MDF apresentou $nx = 201$ pontos nodais, com um intervalo $\Delta x \cong 0,005$ entre eles. No domínio do tempo, os parâmetros da discretização do MDF obedeceram à condição CFL, de acordo com a Equação (2.25). As condições de bordas absorventes, modificadas conforme descrito anteriormente nesta tese, foram empregadas para suprimir reflexões indesejadas nas bordas do modelo.

A Figura 3.10(a) mostra a propagação da onda, ou seja, a evolução do campo de onda $u(x,t)$ em função do tempo t e do espaço x . A Figura 3.10(b) apresenta o traço sísmico registrado pelo receptor. Em ambas as figuras, é possível observar dois eventos distintos: (1) um evento acentuado, de maior amplitude, que representa o sinal da onda que sai da fonte e chega diretamente ao receptor, conhecido como onda direta; e (2) um evento mais atenuado, relacionado à reflexão da onda direta na interface do modelo de velocidade. Além disso, observa-se que o tratamento das condições de bordas absorventes foi altamente preciso, resultando em um campo de onda registrado livre de reflexões indesejadas nas bordas.

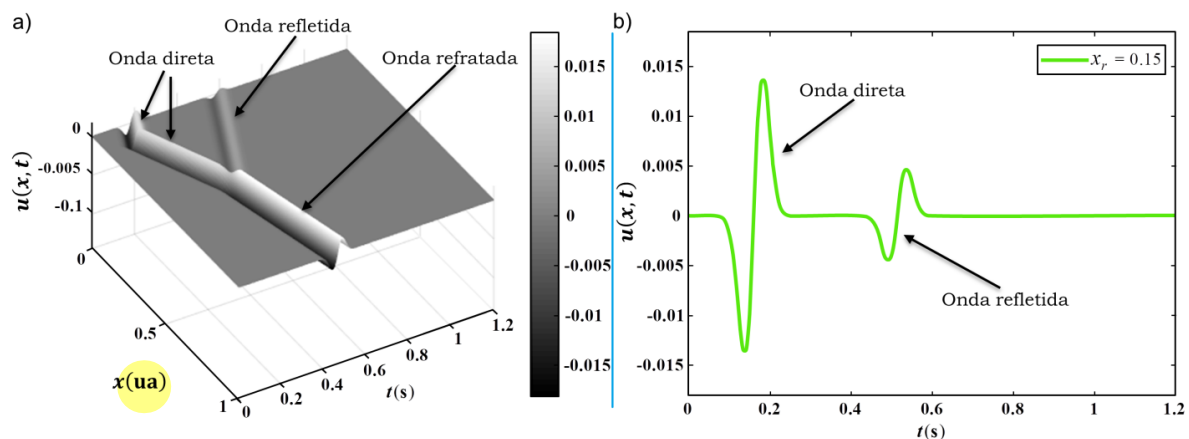


Figura 3.10 – (a) Evolução da propagação do campo de onda $u(x,t)$ e (b) Dados observados \mathbf{d}^{obs} ou traço sísmico sintético registrado em $x_r = 0,15 \text{ du}$.

3.3.3 Parâmetros da otimização híbrida, critérios de parada e recursos Computacionais

Com o intuito de controlar a evolução do algoritmo PSO-Kmeans-ANMS em direção à solução ótima, foram definidos os valores inicial e final para cada parâmetro do PSO, de acordo com a Tabela 3.1. Também foi estabelecido o número máximo de iterações permitidas, It_{max} , para as fases do algoritmo híbrido: PSO na Fase 1 e ANMS na Fase 2. Além disso, foi

determinado o número de *clusters* a serem utilizados pelo algoritmo K-means na divisão do enxame de partículas.

Nesta investigação, durante a Fase 1 do algoritmo, o enxame de partículas foi dividido em dois *clusters* a cada iteração do PSO, utilizando o método K-means. O salto da Fase 1 (PSO) para a Fase 2 (ANMS) ocorreu quando um dos *clusters* se tornou dominante (com quatro vezes mais partículas do que o outro) ou quando o número máximo de iterações, It_{max} foi alcançado. Na Fase 2, o critério de parada para o método ANMS foi definido como um fator de encolhimento simplex $\alpha_s = 10^{-2}$ ou um número máximo de iterações, It_{max} . Vale lembrar que α_s é baseado no desvio padrão dos valores objetivos das soluções no simplex. O K-means só atuou após 50% das iterações permitidas terem ocorrido, resultando em um tempo de salto de $T_{Km} = 0,5 \times It_{max}$. O valor para It_{max} depende da aplicação e foi definido como 54. Além disso, o coeficiente de tamanho do passo do ANMS foi definido como $\beta = 0,5$.

Todos os algoritmos desenvolvidos para o presente trabalho foram implementados utilizando o software MATLAB R2015a. Os experimentos foram simulados em uma máquina com processador Intel(R) Core(TM) i7 e capacidade de memória RAM de 16GB.

sugestão: algoritmos em um apêndice

Capítulo IV

Resultados e Discussões

4 Resultados e Discussões

Neste capítulo, apresenta-se os resultados obtidos pelo algoritmo híbrido PSO-Kmeans-ANMS, utilizado para buscar o mínimo global do problema FWI 1D através da otimização da função de desajuste entre os dados sísmicos observados d^{obs} e calculados d^{calc} . Esses resultados são comparados e discutidos com os obtidos pelos algoritmos PSO classic, PSO mod e ANMS.

4.1 Avaliação do algoritmo híbrido de otimização no FWI

relembrar o que são esses parâmetros

Na modelagem FWI, uma solução é descrita pelos parâmetros de um modelo de velocidade, representado por uma partícula $\mathbf{m} = (m_1 = V_1, m_2 = V_2, m_3 = h_{rf})$. Para avaliar o impacto do tamanho do enxame no processo de inversão FWI, os experimentos foram simulados em três situações distintas: Caso 1, denominado 100×20 , representado por 100 amostras, cada uma composta por 20 modelos iniciais (partículas); Caso 2, denominado 50×40 , correspondente a 50 amostras com 40 modelos iniciais em cada simulação; e Caso 3, 100×10 , com 100 amostras e 10 modelos iniciais em cada simulação. Para gerar aleatoriamente os modelos iniciais nas simulações, utilizou-se a mesma estratégia adotada no experimento numérico com as funções benchmark.

por que?

justificar

Para avaliar como o algoritmo PSO-kmeans-ANMS funciona no problema FWI, foi escolhido o Caso 1 como exemplo. A Figura 4.1 mostra a evolução do algoritmo ao longo de suas iterações. Na Figura 4.1(a), pode-se observar o instante do salto da Fase 1, PSO-Kmeans, para a Fase 2, ANMS. Nessa figura, é perceptível o encolhimento do enxame ao comparar o enxame inicial (círculos verdes) com o final (círculos azuis). Após o PSO-Kmeans-ANMS concluir 50% das iterações permitidas, o K-means passou a atuar na divisão do enxame em dois *clusters* a cada iteração, os quais estão representados na Figura 4.1(b) por dois conjuntos (*clusters* 1 e 2). Na fase 2, o ANMS construiu um simplex inicial quase regular (tetraedro preto) a partir do ponto ótimo (losango amarelo) encontrado pelo PSO na fase anterior. Em seguida, o simplex tetraédrico foi sendo modificado e encolhido pelo algoritmo ANMS, permitindo o seu deslocamento na direção do ótimo global (círculo vermelho), que representa

os parâmetros do modelo verdadeiro de velocidade \mathbf{m} desejado para o problema FWI. Na Figura 4.1(b) verifica-se que o número de partículas do *cluster* 1 superou o do *cluster* 2, com uma diferença de 18 partículas entre eles. A razão entre os tamanhos dos *clusters* foi de $rel = 19/1 \geq rel_s = 4$, dentro do critério previamente estabelecido. Ainda assim, o *cluster* dominante está próximo do ótimo, tornando mais fácil para o ANMS encontrar esse ótimo.

notação
confusa

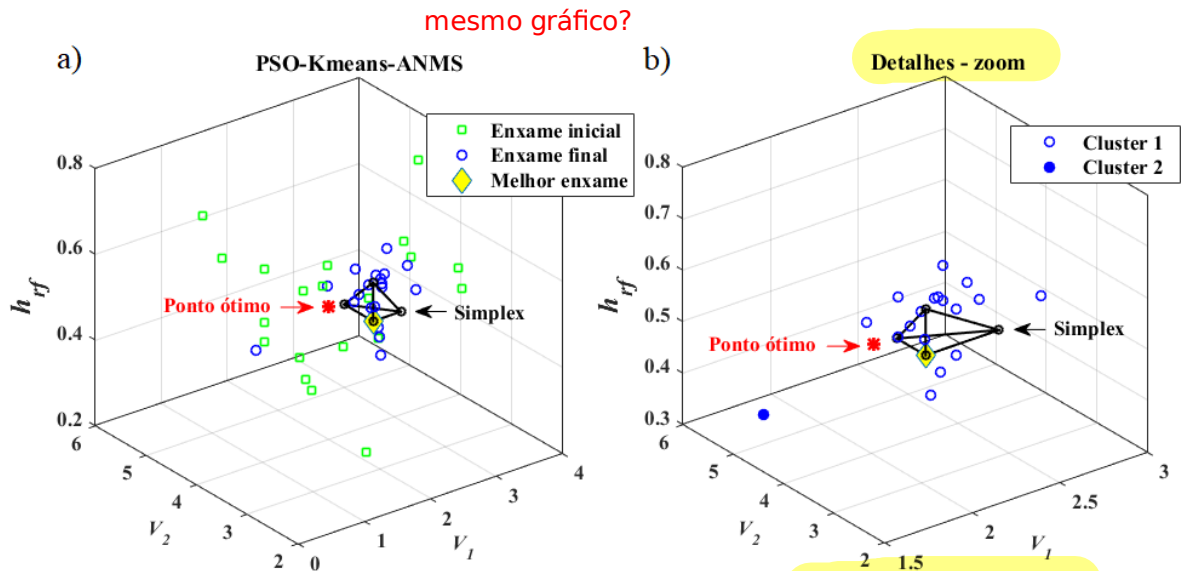


Figura 4.1 – (a) Momento do salto da Fase 1 para a Fase 2; (b) Detalhes do *cluster* 1 (círculos azuis), *cluster* 2 (círculos azuis preenchidos) e da construção do simplex inicial (tetraedro preto).

A Figura 4.2(a), apresenta um gráfico de barras que ilustra o encolhimento do enxame, onde é possível observar que a taxa de encolhimento do enxame foi de 72%, calculada com base no diâmetro do enxame. O diâmetro do enxame foi obtido pela média das distâncias de todas as partículas ao centroide do enxame. A Figura 4.2(b) apresenta os tempos de execução (tempo de CPU) gastos nas Fases 1 e 2, sendo que o tempo de execução da Fase 2 corresponde a aproximadamente 37% do tempo de execução da Fase 1.

especificar máquina

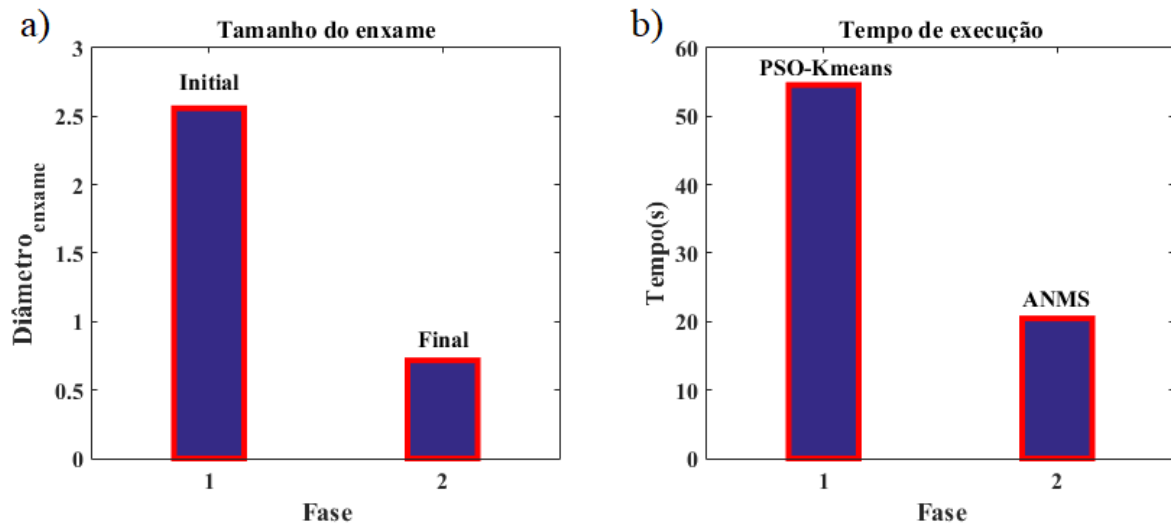


Figura 4.2 – Caso 1, (100×20) : (a) Diâmetros dos enxames inicial e final; (b) Tempo de execução das Fases 1 e 2.

4.2 Avaliação comparativa dos métodos de otimização aplicados na FWI

As informações mais relevantes dos resultados obtidos em cada amostra da simulação computacional foram analisadas e discutidas para os três casos $(100 \times 20; 50 \times 40; 100 \times 10)$. Em todos os casos, foram considerados e comparadas entre si quatro algoritmos de otimização: ANMS, PSO clássico, PSO modificado e PSO-Kmeans-ANMS. Esses algoritmos foram comparados em relação à taxa de sucesso (sr), ao tempo médio de execução (rt) e ao melhor resultado obtido para a função objetivo $\phi(m)$. Em todas as amostras, as soluções encontradas pelos algoritmos de otimização que variaram do modelo real (exato) em um intervalo de $\pm 4\%$ em torno de seus parâmetros (v_1, v_2, h_{rf}) foram aceitas como o ótimo global.

4.2.1 Caso 1

o que define o sucesso?

Primeiramente, foi considerado o Caso 1 com seus resultados apresentados na Tabela 4.1 para os quatro algoritmos de otimização estudados. Observa-se que a abordagem híbrida de otimização obteve a maior taxa de sucesso, ou seja, 100%. Já em relação ao tempo médio de execução, o PSO-Kmeans-ANMS ficou numa posição intermediária entre os outros três algoritmos. Esses resultados eram esperados porque o algoritmo de busca local, ANMS,

converge mais rapidamente do que o algoritmo de busca global PSO, seja ele clássico ou modificado.

Tabela 4.1 – Caso 1, (100×20) : Resultados da taxa de sucesso $sr(\%)$ e do tempo médio de execução $rt(s)$ de cada algoritmo de otimização.

Valores	Algoritmo			
	ANMS	PSO clássico	PSO modificado	PSO-Kmeans-ANMS
$sr(\%)$	48	99	99	100
$rt(s)$	72,3	148	157,2	106,6

A Tabela 4.2 mostra o resultado do melhor modelo $\mathbf{m} = (m_1 = V_1, m_2 = V_2, m_3 = h_{rf})$ obtido para cada algoritmo analisado. Observa-se que a abordagem híbrida de otimização obteve o melhor modelo com o desajuste compatível com os demais algoritmos. As variáveis V_1 , V_2 e h_{rf} parecem ser menos sensíveis ao processo de otimização do que o desajuste.

Tabela 4.2 – Caso 1, (100×20) : Resultados dos parâmetros do modelo e da função objetivo para a melhor solução obtida em cada algoritmo de otimização.

Algoritmo	Parâmetros do modelo			Função objetivo
	$V_1(2,0)$	$V_2(4,0)$	$h_{rf}(0,5)$	$\phi(m) (0,0)$
ANMS	2,0000	4,0000	0,5025	$1,49 \times 10^{-4}$
PSO clássico	2,0000	4,0016	0,5016	$3,98 \times 10^{-3}$
PSO modificado	2,0000	4,0000	0,5043	$6,71 \times 10^{-5}$
PSO-Kmeans-ANMS	2,0000	4,0000	0,5006	$1,28 \times 10^{-4}$

A Tabela 4.3 mostra os valores médios obtidos e os seus respectivos desvios padrão para cada parâmetro (V_1, V_2, h_{rf}) do modelo. Já na Tabela 4.4, estão os valores médios do tempo de execução da CPU e da função objetivo, *misfit* $\phi(m)$, obtidos por cada algoritmo. Nesta avaliação, apenas os parâmetros dos modelos bem-sucedidos foram computados. Observa-se que todos os algoritmos de otimização obtiveram resultados semelhantes e bastante satisfatórios para os parâmetros do modelo e para o valor da função objetivo. Quanto

ao tempo de CPU, o algoritmo híbrido PSO-Kmeans-ANMS obteve uma redução aproximada de 28% em relação ao PSO clássico e de 32% em relação ao PSO modificado. Isso representa um ganho significativo quando se trata de um problema de otimização com uma função objetivo de alto custo computacional.

Tabela 4.3 – Caso 1, (100×20) : Resultados do valor médio e do respectivo desvio padrão para os parâmetros dos modelos bem-sucedidos.

Algoritmo	Parâmetros do modelo		
	$V_1(2,0)$	$V_2(4,0)$	$h_{rf}(0,5)$
ANMS	$2,01 \pm 4,08 \times 10^{-2}$	$4,01 \pm 5,73 \times 10^{-2}$	$0,504 \pm 8,03 \times 10^{-3}$
PSO clássico	$2,00 \pm 9,34 \times 10^{-3}$	$4,00 \pm 3,35 \times 10^{-2}$	$0,502 \pm 2,31 \times 10^{-3}$
PSO modificado	$2,00 \pm 1,40 \times 10^{-2}$	$4,00 \pm 2,48 \times 10^{-2}$	$0,503 \pm 3,37 \times 10^{-3}$
PSO-Kmeans-ANMS	$2,01 \pm 1,96 \times 10^{-2}$	$4,01 \pm 2,57 \times 10^{-2}$	$0,504 \pm 4,12 \times 10^{-3}$

Tabela 4.4 – Caso 1, (100×20) : Resultados do valor médio e do respectivo desvio padrão para o tempo de CPU e a função objetivo.

Algoritmo	Tempo de execução	Função objetivo
ANMS	$72,30 \pm 17,18$	$0,442 \pm 4,00 \times 10^{-1}$
PSO clássico	$147,99 \pm 10,15$	$0,111 \pm 1,34 \times 10^{-1}$
PSO modificado	$157,20 \pm 16,45$	$0,124 \pm 1,69 \times 10^{-1}$
PSO-Kmeans-ANMS	$106,58 \pm 11,80$	$0,194 \pm 2,28 \times 10^{-1}$

não foi o menor, deixar claro que isso não é tão relevante

As Figuras 4.3(a), 4.3(b) e 4.3(c) mostram os resultados para os parâmetros dos modelos de velocidade calculados pelo algoritmo de otimização ANMS, distribuídos em torno dos valores exatos $V_1 = 2,0$, $V_2 = 4,0$ e $h_{rf} = 0,5$, respectivamente. Os círculos azuis representam os modelos bem-sucedidos, com valores dentro do intervalo definido para a solução ótima, enquanto os círculos vermelhos indicam os modelos que não foram bem-sucedidos. As Figuras 4.3(d), 4.3(e) e 4.3(f) mostram os histogramas correspondentes aos círculos azuis para cada parâmetro do modelo. Os histogramas demonstram que o ANMS testou muitos modelos para cada espaço de parâmetros e que os modelos aceitos se agrupam perto do modelo verdadeiro.

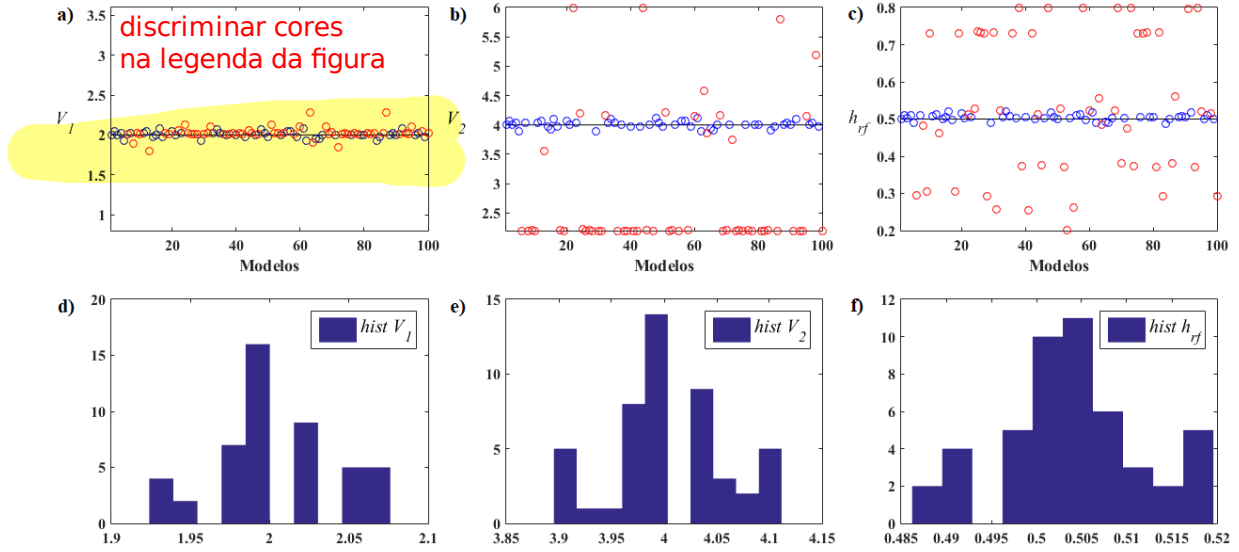


Figura 4.3 – Caso 1, (100×20) : (a), (b) e (c) todos os resultados do algoritmo ANMS para as variáveis V_1 , V_2 e h_{rf} , respectivamente. (d), (e) e (f) histogramas correspondentes ao modelos bem-sucedidos para cada variável do modelo.

Analogamente, as Figuras 4.4, 4.5 e 4.6 mostram os resultados para os parâmetros dos modelos de velocidade calculados para os demais algoritmos de otimização utilizados neste estudo: PSO clássico, PSO modificado e PSO-Kmeans-ANMS, respectivamente.

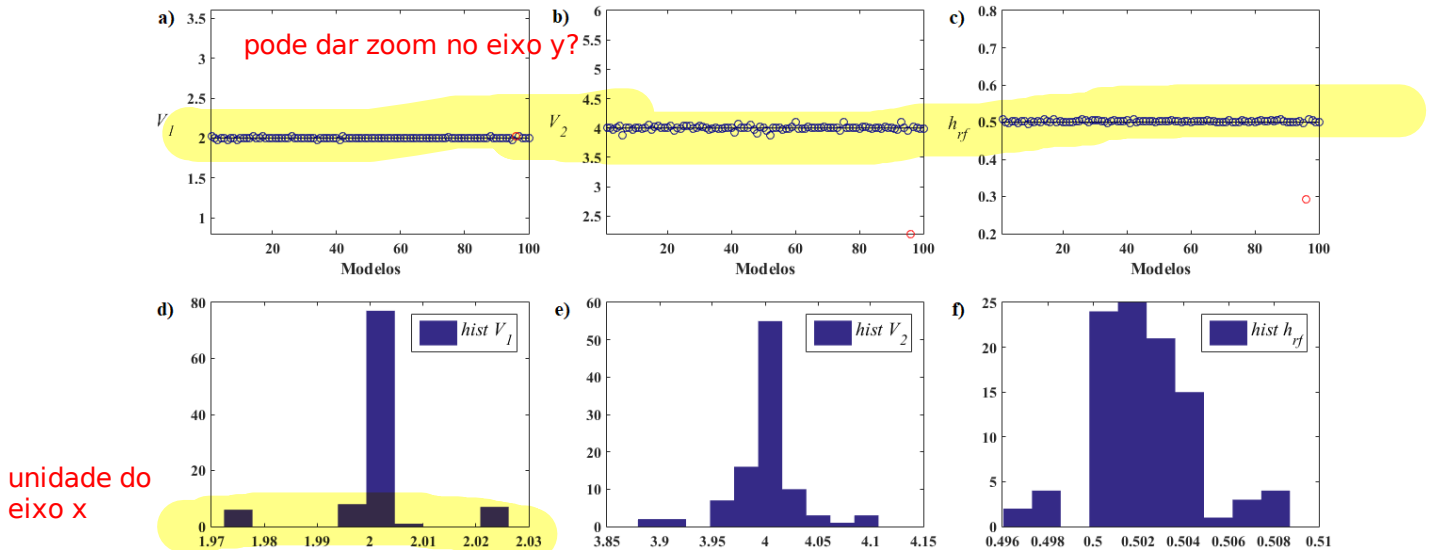


Figura 4.4 – Caso 1, (100×20) : (a), (b) e (c) todos os resultados do algoritmo PSO clássico para as variáveis V_1 , V_2 e h_{rf} , respectivamente. (d), (e) e (f) histogramas correspondentes ao modelos bem-sucedidos para cada variável do modelo.

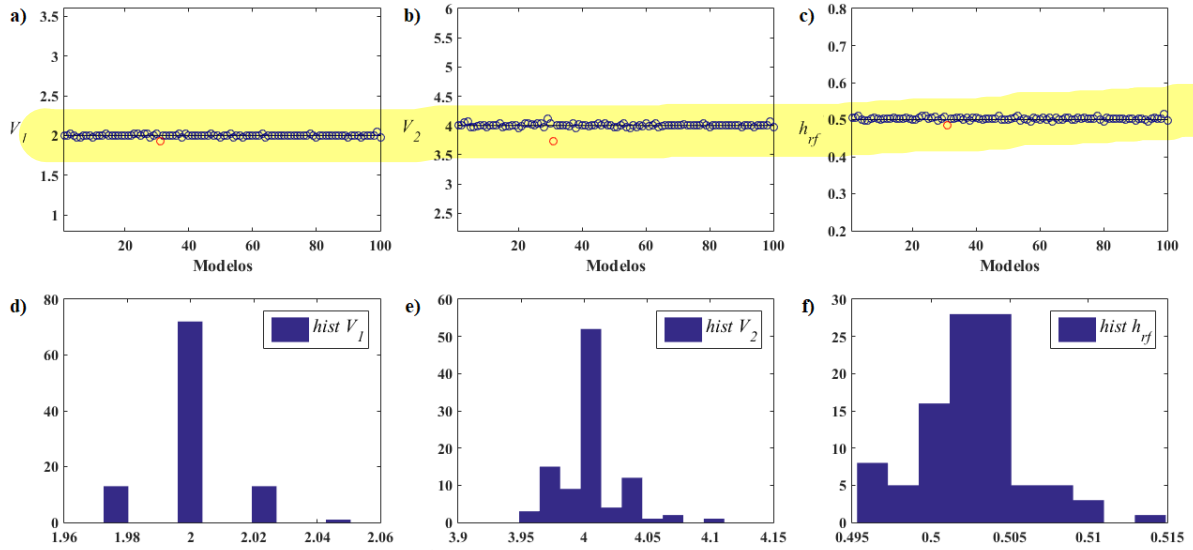


Figura 4.5 – Caso 1, (100×20) : (a), (b) e (c) todos os resultados do algoritmo PSO modificado para as variáveis V_1 , V_2 e h_{rf} , respectivamente. (d), (e) e (f) histogramas correspondentes ao modelos bem-sucedidos para cada variável do modelo.

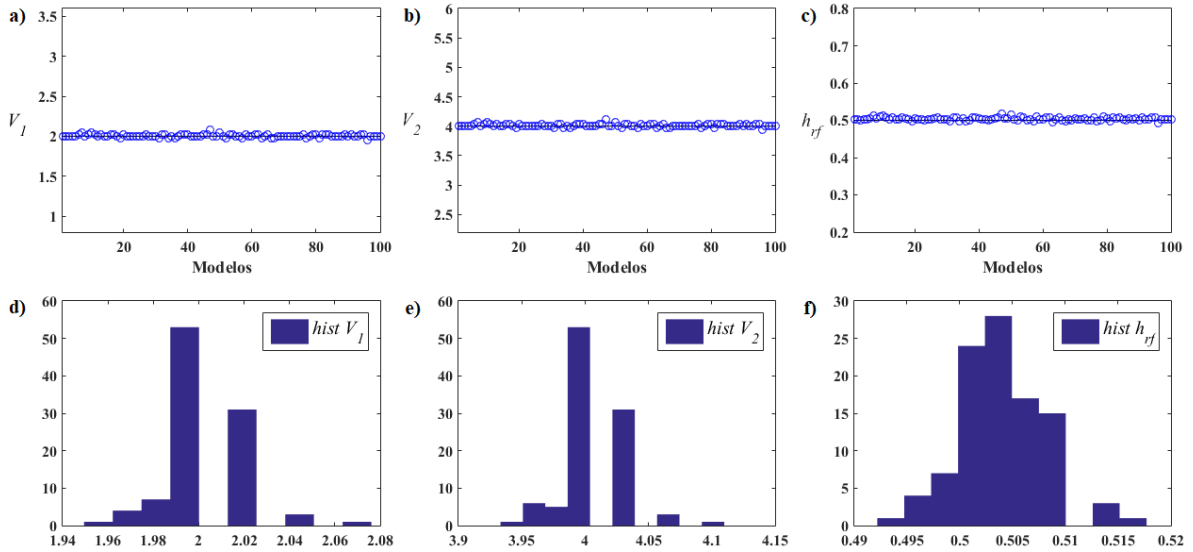


Figura 4.6 – Caso 1, (100×20) : (a), (b) e (c) todos os resultados do algoritmo PSO-Kmeans-ANMS para as variáveis V_1 , V_2 e h_{rf} , respectivamente. (d), (e) e (f) histogramas correspondentes ao modelos bem-sucedidos para cada variável do modelo.

Na sequência, a Figura 4.7 apresenta o histograma do tempo de execução para cada abordagem de otimização. Todos os histogramas foram construídos com base na avaliação dos modelos bem-sucedidos. Neles, é possível visualizar a dispersão dos valores em relação ao valor médio do tempo de execução, que foi mostrado na Tabela 4.4. Observa-se, na Figura

4.7(b) que os modelos bem-sucedidos obtidos com o PSO clássico ficaram mais concentrados em torno do valor médio do tempo de execução.

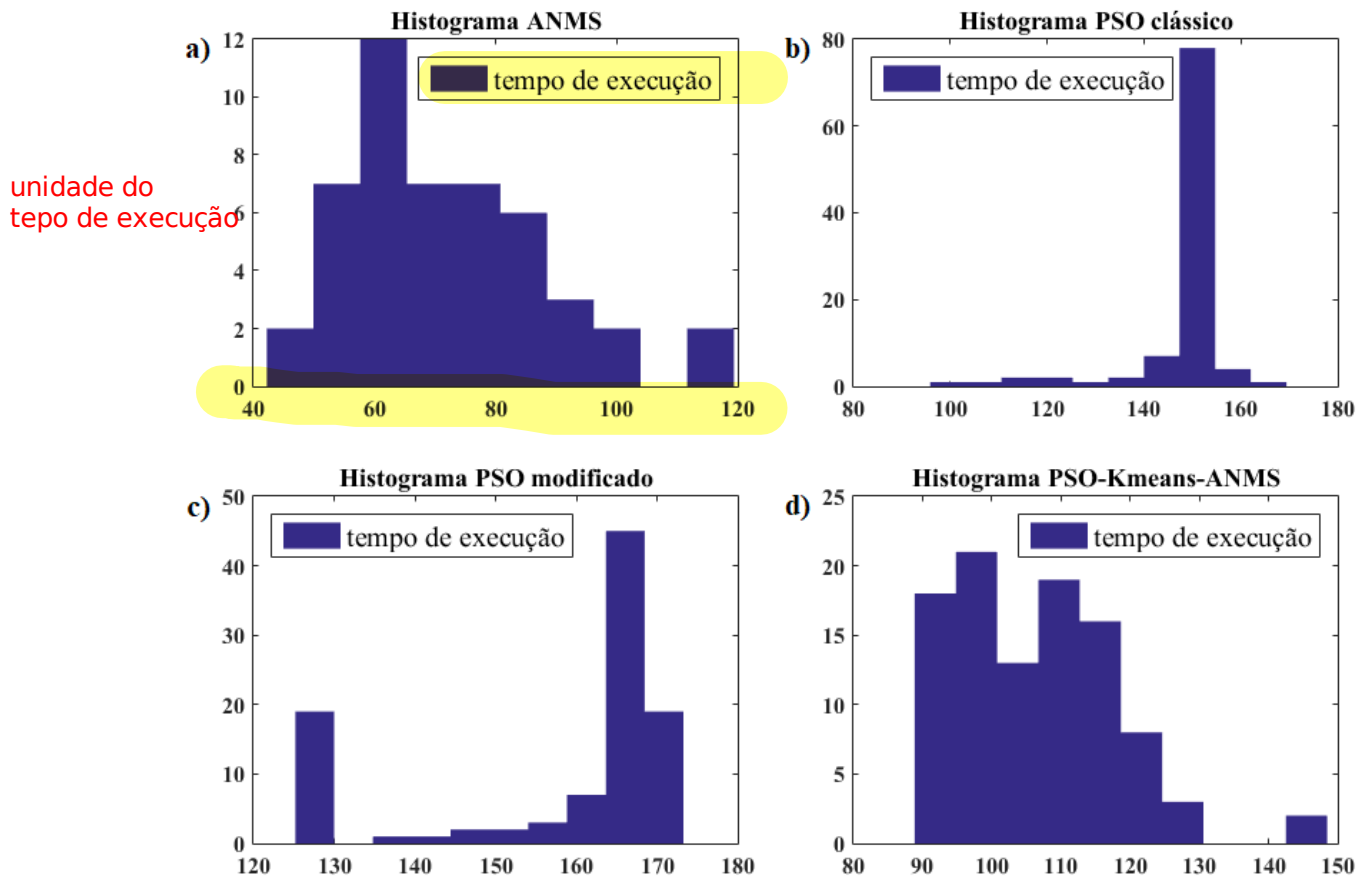


Figura 4.7 – Caso 1, (100×20) : Histograma do tempo de CPU para cada algoritmo de otimização, aplicados apenas aos modelos bem-sucedidos.

A Figura 4.8(a) apresenta um resumo gráfico referente ao Caso 1 aplicado ao FWI. Esse resumo, com os resultados obtidos nas simulações correspondentes a cada abordagem de otimização, foi apresentado em classes de valor. Sendo a Classe 1 referente à taxa de sucesso (%); a Classe 2, à média dos tempos de execução (em segundos) e a Classe 3, à média dos desajustes $(\phi(m) \times 10^2)$. A Figura 4.8(b) mostra os valores obtidos da função objetivo apenas para os modelos bem-sucedidos.

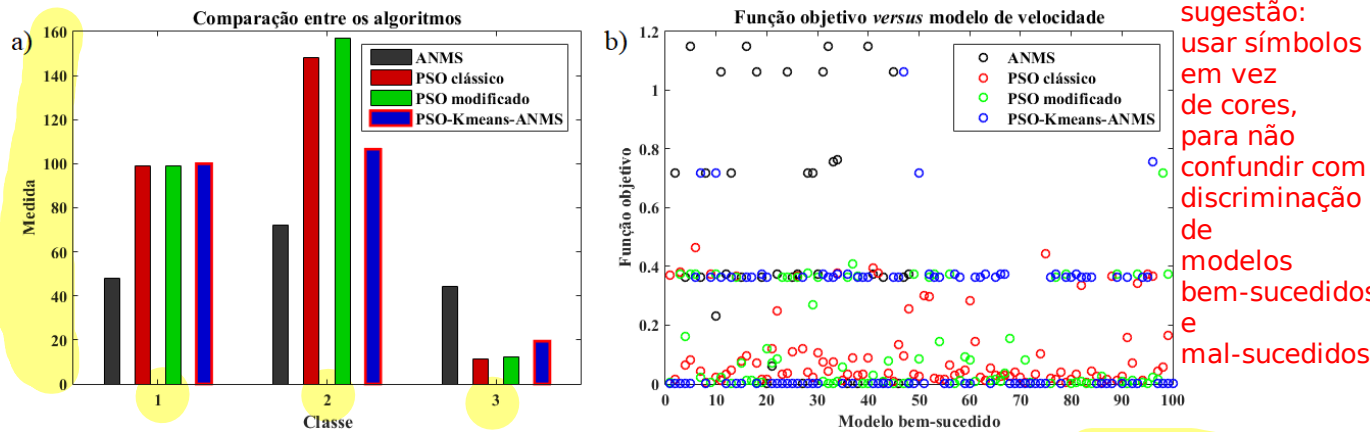


Figura 4.8 – Caso 1, (100×20) : (a) Comparação entre os algoritmos divididos em: Classe 1, taxas de sucesso; Classe 2, tempos médios de CPU e Classe 3, valores médios da função objetivo $[\phi(m) \times 10^2]$. (b) Valores da função objetivo para os modelos bem-sucedidos.

confuso

Apartir das Figuras 4.3, 4.4, 4.5 e 4.6, pode-se afirmar que os modelos bem-sucedidos, em azul, estão mais concentrados em torno dos valores exatos $V_1 = 2,0$, $V_2 = 4,0$ e $h_{rf} = 0,5$, porém em menor grau para o algoritmo ANMS. Tal comportamento ocorre porque o ANMS é um otimizador local, enquanto o PSO é um otimizador global que aumenta a chance de escapar de mínimos locais. Quanto aos histogramas, todos apresentaram dispersões pouco significativas em torno da média.

Na Figura 4.8(a), é possível observar que o PSO-Kmeans-ANMS obteve uma taxa de sucesso de 100% e conseguiu reduzir o tempo de CPU com a diminuição da duração de atuação do PSO. Dessa forma, atingiu um valor bastante aceitável para função objetivo. Na Figura 4.8(b), temos uma visão mais detalhada dos valores desses desajustes para os modelos bem-sucedidos. De acordo com os dados, pode-se constatar que todos os algoritmos apresentaram valores próximos ao valor mínimo da função objetivo.

4.2.2 Caso 2

As simulações para o Caso 2 (50×40) foram avaliadas, extraindo os dados relevantes para comparar com o Caso 1 (100×20) . Condições semelhantes às do Caso 1 foram aplicadas, com exceção do tamanho do enxame, que possui o dobro das partículas, e da redução pela metade do número de simulações. Na Tabela 4.5, apresentam-se a taxa de

sucesso, sr ; e o tempo médio de execução, rt , para cada abordagem de otimização utilizada nesta pesquisa.

Tabela 4.5 – Caso 2, (50×40) : Resultados da taxa de sucesso $sr(\%)$ e do tempo médio de execução $rt(s)$ de cada algoritmo de otimização.

Valores	Algoritmo			
	ANMS	PSO clássico	PSO modificado	PSO-Kmeans-ANMS
$sr(\%)$	62	100	100	100
$rt(s)$	66,9	231	227,3	166,4

Conforme mostrado na Tabela 4.5 referente ao Caso 2 (50×40) , todos os algoritmos de otimização obtiveram uma taxa de sucesso de 100%, exceto o algoritmo ANMS. A abordagem híbrida de otimização, PSO-Kmeans-ANMS, manteve sua taxa de sucesso e as demais abordagens houve aumento dos seus valores em comparação com o Caso 1. No que se refere ao tempo de execução, o PSO-Kmeans-ANMS ficou em uma posição intermediária entre os demais algoritmos. O algoritmo híbrido manteve a redução aproximada de 28% em relação ao PSO clássico, enquanto o PSO modificado apresentou um decréscimo de 32% no Caso 1 para 27% no Caso 2. A Tabela 4.6 mostra os melhores resultados alcançados por cada abordagem de otimização. Nota-se mudanças pouco significativas nos resultados obtidos para os parâmetros do modelo. Também observa-se que com um enxame maior as quatro abordagens de otimização obtiveram melhores resultados. Isso parece ser intuitivo, ou seja, quanto mais partículas, maior é a chance do algoritmo convergir para uma bacia de atração. No entanto, o tempo de execução cresce significativamente devido ao aumento do número de avaliações da função objetivo. Isso torna o experimento das simulações mais dispendioso computacionalmente.

Tabela 4.6 – Caso 2, (50×40) : Resultados dos parâmetros do modelo e da função objetivo para a melhor solução obtida em cada algoritmo de otimização.

Algoritmo	Parâmetros do modelo			Função objetivo
	$V_1(2,0)$	$V_2(4,0)$	$h_{rf}(0,5)$	$\phi(m)(0,0)$
ANMS	2,0000	4,0000	0,5025	$1,49 \times 10^{-4}$
PSO clássico	2,0000	4,0006	0,5017	$1,83 \times 10^{-3}$
PSO modificao	2,0000	4,0000	0,5020	$3,66 \times 10^{-5}$
PSO-Kmeans-ANMS	2,0000	4,0000	0,5018	$1,33 \times 10^{-4}$

Por fim, a Figura 4.9(a) mostra o resumo gráfico referente ao Caso 2 (50×40) para a técnica FWI. Nesse resumo, cada abordagem de otimização foi agrupada em uma classe de valor para facilitar a comparação entre os resultados obtidos. A Figura 4.9(b) apresenta os valores obtidos pela função objetivo apenas para os modelos bem- sucedidos.

Além disso, com base na Figura 4.9(a), semelhante ao Caso 1 (100×20) , o algoritmo híbrido de otimização, PSO-Kmeans-ANMS, manteve a taxa de sucesso de 100% com o valor da função objetivo bastante próximo do mínimo exato. Porém, não foi capaz de reduzir o tempo de CPU. Na Figura 4.9(b), embora todos os valores dos desajustes tenham sido mantidos no mesmo intervalo de valores, o algoritmo PSO (clássico ou modificado) apresentou uma distribuição mais uniforme e ficou bastante próximo do valor exato. De modo geral, houve um ganho quando o tamanho do enxame foi dobrado, principalmente para o PSO (clássico e modificado). No entanto, houve um certo custo dado pelo aumento significativo no tempo de execução dos algoritmos de otimização.

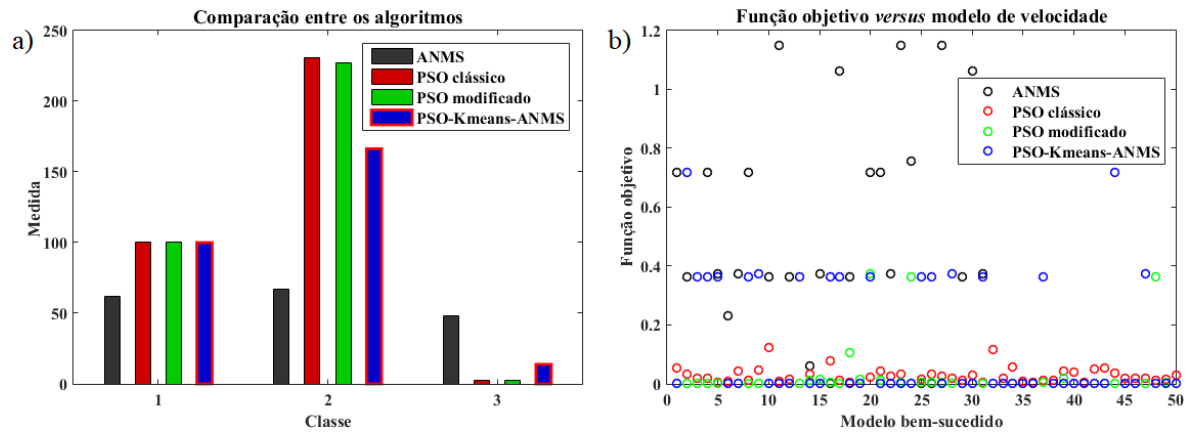


Figura 4.9 – Caso 2, (50×40) : (a) Comparação entre os algoritmos divididos em: Classe 1, taxas de sucesso; Classe 2, tempos médios de CPU e Classe 3, valores médios da função objetivo $[\phi(m) \times 10^2]$. (b) Valores da função objetivo para os modelos bem-sucedidos.

4.2.1 Caso 3

As simulações para o Caso 3 (100×10) foram analisadas, extraindo os dados relevantes para comparar com o Caso 1 (100×20) . Novamente, foram adotadas condições semelhantes às do Caso 1. A única diferença está no tamanho do enxame que tem metade das partículas do Caso 1. A Tabela 4.7, apresenta os resultados da taxa de sucesso e do tempo médio de execução para cada abordagem de otimização utilizada nesta pesquisa.

Tabela 4.7 – Caso 3, (100×10) : Resultados da taxa de sucesso $sr(\%)$ e do tempo médio de execução $rt(s)$ de cada algoritmo de otimização.

Valores	Algoritmo			
	ANMS	PSO clássico	PSO modificado	PSO-Kmeans-ANMS
$sr(\%)$	40	77	76	96
$rt(s)$	73	56,6	56,2	77,5

Conforme mostrado na Tabela 4.7, para o Caso 3 (100×10) , todos os algoritmos de otimização tiveram uma redução na taxa de sucesso. No entanto, o PSO-Kmeans-ANMS foi o que apresentou a menor redução, mantendo-se assim na liderança. Em relação ao tempo de execução, todos os algoritmos de otimização apresentaram uma redução considerável, exceto

o ANMS, que praticamente manteve seu valor. Nesse quesito, o desempenho do algoritmo híbrido, PSO-Kmeans-ANMS, na fase 2 ficou prejudicado devido à sua dependência em relação ao ANMS.

Portanto, com menos partículas no enxame, o algoritmo híbrido conseguiu se manter em uma situação bastante confortável com relação à taxa de sucesso, embora tenha perdido em tempo de CPU. Ou seja, houve um comprometimento entre a redução do tamanho do enxame e o tempo de CPU.

A Tabela 4.8 mostra a melhor solução obtida por cada abordagem de otimização. Observa-se mudanças pouco significativas nos resultados dos parâmetros do modelo, em comparação ao Caso 1.

Tabela 4.8 – Caso 3, (100×10) : Resultados dos parâmetros do modelo e da função objetivo para a melhor solução obtida em cada algoritmo de otimização.

Algoritmo	Parâmetros do modelo			Função objetivo
	$V_1(2,0)$	$V_2(4,0)$	$h_{rf}(0,5)$	$\phi(m)(0,0)$
ANMS	2,0000	4,0000	0,5025	$1,49 \times 10^{-4}$
PSO clássico	1,9999	4,0022	0,5043	$7,38 \times 10^{-3}$
PSO modificao	2,0000	4,0002	0,5013	$1,07 \times 10^{-3}$
PSO-Kmeans-ANMS	2,0000	4,0000	0,5022	$9,86 \times 10^{-5}$

Finalmente, a Figura 4.10(a) apresenta o resumo gráfico relativo ao Caso 3 (100×10) para a FWI. Nesse resumo cada abordagem de otimização foi agrupada em uma classe de valor. A Figura 4.10(b) mostra os valores obtidos pela função objetivo somente para os modelos bem-sucedidos.

A Figura 4.10(a) mostra que o algoritmo híbrido de otimização PSO-Kmeans-ANMS manteve uma taxa de sucesso maior em comparação com os demais algoritmos do Caso 3, com o valor da função objetivo bastante próximo do mínimo exato, apesar de não ter conseguido reduzir o tempo de CPU. Na Figura 4.10(b), os algoritmos apresentaram uma distribuição menos uniforme e ficaram muito próximos do valor exato, embora todos os

valores de desajuste tenham permanecido no mesmo intervalo de valores. Em relação ao tempo de CPU para o PSO (clássico ou modificado), houve um ganho quando o tamanho do enxame foi reduzido pela metade, tendo como consequência uma diminuição significativa em suas taxas de sucesso.

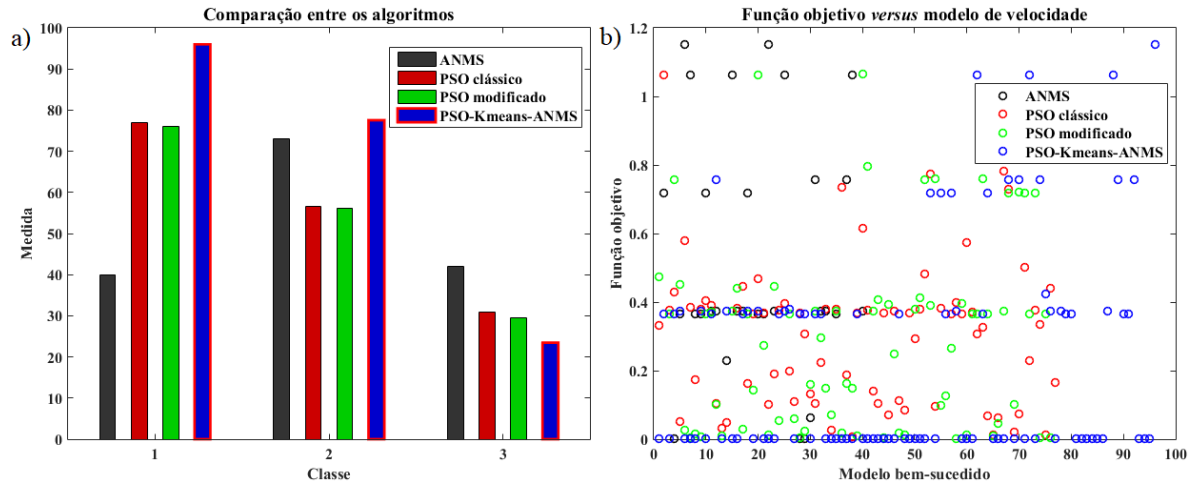


Figura 4.10 – Caso 3, (100×10) : (a) Comparação entre os algoritmos divididos em: Classe 1, taxas de sucesso; Classe 2, tempos médios de CPU e Classe 3, valores médios da função objetivo $[\phi(m) \times 10^2]$. (b) Valores da função objetivo para os modelos bem-sucedidos.

A seguir, apresenta-se um resumo dos dados obtidos para os três casos estudados nesta pesquisa, em termos de taxa de sucesso, $sr(\%)$, e de tempo médio de execução, $rt(s)$, para os algoritmos de otimização utilizados. Para isso, a Figura 4.11 mostra uma visualização gráfica dos dados apresentados nas Tabelas 4.1, 4.5 e 4.7, referentes ao Caso 1 (100×20) com 20 partículas, ao Caso 2 (50×40) com 40 partículas e ao Caso 3 (100×10) com 10 partículas, respectivamente. Os casos são identificados pelo número de partículas do enxame.

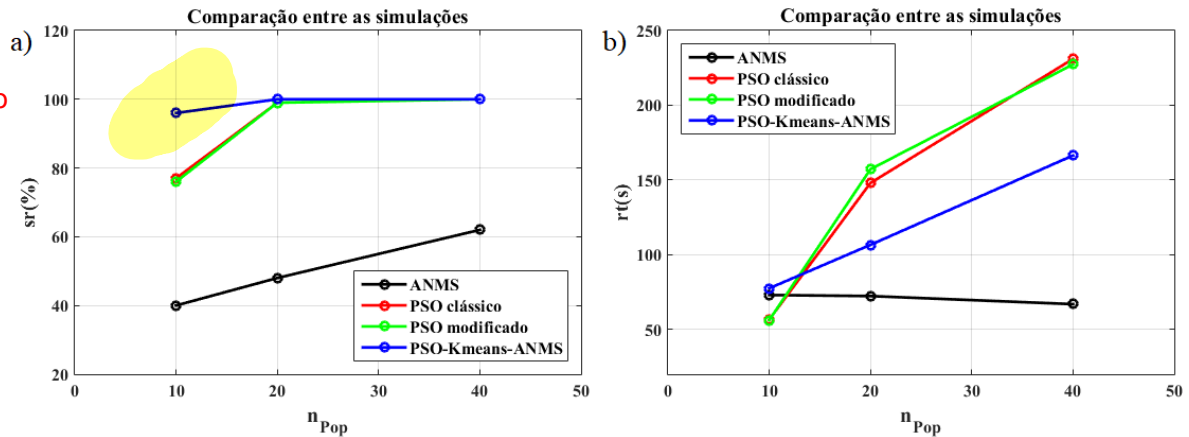


Figura 4.11 – Comparação entre as simulações para todos os algoritmos e Casos: (a) taxa de sucesso *versus* população e (b) tempo médio de execução *versus* população.

A partir da Figura 4.11(a), pode-se observar que o algoritmo ANMS apresenta um comportamento crescente e, aproximadamente, linear para a taxa de sucesso, $sr(\%)$, em relação ao tamanho do enxame, n_{Pop} . Já os demais algoritmos apresentaram um comportamento crescente assintótico. Na Figura 4.11(b), os algoritmos PSO clássico, PSO modificado e PSO-kmeans-ANMS mantiveram um comportamento crescente, sendo que este último apresentou um comportamento aproximadamente linear. O algoritmo ANMS manteve seu comportamento linear, mas praticamente constante, para o tempo de execução, $rt(s)$, em relação ao tamanho do enxame. Portanto, com base nestes gráficos, pode-se concluir que o algoritmo híbrido de otimização PSO-Kmeans-ANMS apresenta um comportamento crescente e assintótico para a $sr(\%)$ e um comportamento linear crescente para o $rt(s)$. Observa-se também que uma população de enxame igual a 20 constitui uma situação de equilíbrio para o seu comportamento, ou seja, neste ponto, o algoritmo tem o seu melhor desempenho.

Capítulo V

Conclusões e Recomendações

5 Conclusões e Perspectivas futuras

Neste capítulo estão descritas as principais conclusões obtidas nesta pesquisa e algumas perspectivas que poderão ser empregadas em trabalhos futuros.

5.1 Conclusões

No presente trabalho, foi desenvolvido um novo algoritmo híbrido de otimização de duas fases baseado em Otimização Livre de Derivadas (DFO) e em técnica de agrupamento, denominado algoritmo PSO-Kmeans-ANMS. Na Fase 1, uma versão modificada do otimizador global Particle Swarm Optimization (PSO) é usada e, a cada iteração, o algoritmo de agrupamento K-means divide o enxame em dois grupos. Quando um dos grupos se torna dominante, a Fase 1 termina. Na Fase 2, a abordagem de otimização proposta faz uso de um otimizador local, o algoritmo de Nelder-Mead (ANMS). A curva de Hilbert é usada na Fase 1 para linearizar o espaço de busca da solução ótima e dividi-lo em regiões iguais. Isso permite gerar aleatoriamente a população inicial do PSO com maior diversidade para o enxame de forma eficiente e fácil. O conceito de parâmetros adaptativos juntamente com o uso de partículas rebeldes foi amplamente empregado, oferecendo maior dinâmica ao algoritmo PSO. Usando o mesmo grupo de enxames de partículas, o algoritmo híbrido de otimização foi comparado com cada um dos algoritmos ANMS, PSO clássico e PSO modificado, para avaliar seu desempenho em termos de precisão, eficiência e robustez. O algoritmo proposto foi validado através da minimização de 12 funções de referência 2D, dos mais variados tipos e complexidades e, então, aplicado a um problema de *Full Waveform Inversion* (FWI). Matematicamente, o FWI é um problema de otimização, cuja função objetivo requer a solução da equação da onda sísmica em uma dimensão. Resolvemos a modelagem direta da equação de onda pelo Método das Diferenças Finitas (FDM) usando as condições de bordas absorventes que tornam os dados livres de reflexões espúrias nos limites do domínio. A técnica FWI é um problema de alto custo computacional e de difícil solução.

De maneira geral, o algoritmo PSO-Kmeans-ANMS se mostrou bastante eficiente e preciso quando comparado às outras técnicas aqui utilizadas. A sua robustez foi garantida por ter conseguido resolver problemas dos mais variados tipos. A principal contribuição em

termos práticos desta pesquisa, contudo, foi ter reduzido o tempo de computação para o problema FWI. com relação ao PSO clássico e modificado, alcançando uma precisão bem melhor que a do ANMS.

5.2 Perspectivas Futuras

Os resultados obtidos nesta tese suscitam vários questionamentos, principalmente sobre possíveis melhorias e novas aplicações. No que diz respeito ao primeiro ponto, a restrição dos valores que os parâmetros do modelo podem assumir é promissora, pois restringir apenas os valores inteiros deve acelerar o processo de inversão sem causar grandes prejuízos à precisão dos resultados. Além disso, utilizar apenas a representação binária dos valores é um tipo de otimização que tem ganhado muita relevância atualmente. No entanto, a aplicação em problemas de maiores dimensões é obviamente mais relevante, devido às características das aplicações reais da técnica FWI. Como o objetivo desta pesquisa foi apresentar uma nova metodologia híbrida e estudar para baixas dimensões, com poucos parâmetros, dado que a maioria dos métodos DFO apresenta redução na precisão e eficiência à medida que o número de parâmetros aumenta, a aplicação direta da metodologia aqui apresentada deve ficar restrita a problemas reais de inversão de impedância e problemas similares. A metodologia desenvolvida neste trabalho pode ser aplicada diretamente ao FWI 1.5D para casos de simples e média complexidade. Para a FWI 2D, é indicada somente para casos simples, como realizado por Gomes et al. (2017). Novos avanços na parametrização eficiente dos campos de propriedades (velocidades P e S, porosidade, etc.) permitiriam a escalabilidade para aplicações reais mais complexas. A redução drástica no número de parâmetros do modelo permitiria a aplicação da metodologia à técnica FWI em diversas dimensões, como 2D, 2.5D e 3D, assim como em modelos mais complexos, como os que utilizam modelagem viscoelástica. Isso ocorre porque a FWI tradicional, com otimização local pelo método adjunto e semelhantes, torna-se proibitivo computacionalmente para essa modelagem.

será mesmo?

Terminar seu trabalho com outra frase mais otimista.

Referências Bibliográficas

Referências Bibliográficas

- Abualigah, L.; Diabat, A.; Mirjalili, S.; Elaziz, M.A.; Gandomi, A.H. *The Arithmetic Optimization Algorithm*. Computer methods in applied mechanics and engineering, 2021, vol. 376, n.113609, pp. 1-38. doi:10.1016/j.cma.2020.113609
- Aguiar, R.; Neto, A.B.; Bezerra, Y.S.F.; Henriques, M.V.C.; Nascimento, H.A.D.; Lucena, L.S. *A Global Optimization DFO-CRS Strategy for 1D Full Waveform Inversion*. Journal of Computational Interdisciplinary Sciences, 2019, vol. 10, n.1, pp. 21-30. doi: 10.6062/jcis.2019.10.01.0156
- Aleardi, M.; Mazzotti, A. *1D elastic full-waveform inversion and uncertainty estimation by means of a hybrid genetic algorithm-Gibbs sampler approach*. Geophysical Prospecting, 2016, vol. 65, n.1, pp. 64-85. Available from: <https://doi.org/10.1111/1365-2478.12397>.
- Alford, R.M.; Kelly, K.R.; Boore, D.M. *Accuracy of finite-difference modeling of the acoustic wave equation*. Geophysics, 1974; vol. 39, n.6, pp. 834-842. doi:10.1190/1.1440470.
- Araújo, R.R.F.; Gross, L.; de Souza, S.X. *Boosting memory access locality of the Spectral Element Method with Hilbert space-filling curves*. Computers & Geosciences, 2021, vol. 157, pp. 1-23. doi:10.1016/j.cageo.2021.104938.
- Bleistein, N.; Cohen, J.K.; Stockwell Jr., J.W. *Mathematics of Multidimensional Seismic Imaging, Migration, and Inversion*. Springer, New York; 2001. Available from: <https://doi.org/10.1007/978-1-4613-0001-4>.
- Brossier, R.; Operto, S.; Virieux, J. *Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion*. Geophysics, 2009, vol. 74, n.6, pp. WCC105-WCC118. doi:10.1190/1.3215771

- Carlisle, A.; Dozier, G. *An off-the-shelf PSO*. In: Proceedings of the Workshop on Particle Swarm Optimization, 2001, pp. 1-6.
- Cataldi, G.; Abedi, A.; Magnifico, G.; Notarnicola, S.; Dalla Pozza, N.; Giovannetti, V.; Montangero, S. *Hilbert curve vs Hilbert space: exploiting fractal 2D covering to increase tensor network efficiency*. Quantum, 2021, vol. 5, pp. 556. doi: 10.22331/q-2021-09-29-556
- Conn, A. R.; Scheinberg, K.; Vicente, L.N. *Introduction to derivative-free optimization*. Society for Industrial and Applied Mathematics, Philadelphia, 2009. Available from: <https://doi.org/10.1137/1.9780898718768>.
- Chen, G.; Wang, Z. *Robust full-waveform inversion based on particle swarm optimization*. In: SEG Technical Program Expanded Abstracts 2017. Society of Exploration Geophysicists, 2017, pp. 1302-1306. Available from: <https://doi.org/10.1190/segam2017-17634273.1>.
- Chen, S.; Montgomery, J.; Bolufé-Röhler, A. *Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution*. Applied Intelligence, 2014, vol. 42, n.3, pp. 514-526. doi:10.1007/s10489-014-0613-2
- Curtis, A.; Lomax, A. *Prior information, sampling distributions, and the curse of dimensionality*. Geophysics, 2001, vol. 66, n.2, pp. 372-378. doi:10.1190/1.1444928.
- Clerc, M. *The swarm and the queen: Towards a deterministic and Adaptive Particle Swarm Optimization*. In: Proceedings of the IEEE Congress on Evolutionary Computation, 1999, pp. 1951-1957. doi:10.1109/cec.1999.785513
- Clerc, M.; Kennedy, J. *The Particle Swarm - Explosion, stability, and convergence in a multidimensional complex space*. IEEE Transactions on Evolutionary Computation, 2002, vol. 6, n.1, pp. 58-73. doi:10.1109/4235.985692.

- Dai, Y.; Liu, L.; Li, Y. *An intelligent parameter selection method for Particle Swarm Optimization algorithm*. In: 2011 Fourth International Joint Conference on Computational Sciences and Optimization, 2011, pp. 960-964. doi:10.1109/cso.2011.79.
- Datta, D.; Sen, M.K. *Estimating a starting model for full-waveform inversion using a global optimization method*. Geophysics, 2016, vol. 81, n.4, pp. R211–R223. doi:10.1190/geo2015-0339.1.
- Datta, D.; Sen, M.K.; Liu, F.; Morton, S. *Full-waveform inversion of salt models using shape optimization and simulated annealing*. Geophysics, 2019, vol. 84, n.5, pp. R793-R804. doi:10.1190/geo2018-0175.1.
- de Freitas, J.E.; de Lima, R.F.; dos Santos, D.T. *The n-dimensional Peano curve*, São Paulo Journal of Mathematical Sciences, 2019, vol. 13, n.2, pp. 678-688. doi:10.1007/s40863-019-00132-9.
- Diouane, Y.; Gratton, S.; Vasseur, X.; Vicente, L.N.; Calandra, H. *A parallel evolution strategy for an earth imaging problem in geophysics*. Optimization and Engineering, 2016, vol. 17, n.1, pp. 3-26. doi:10.1007/s11081-015-9296-8.
- Eberhart, R.C.; Shi, Y. *Comparing inertia weights and constriction factors in Particle Swarm Optimization*. In: Proceedings of the 2000 Congress on Evolutionary Computation, 2000, vol. 1, pp. 84-99. doi:10.1109/cec.2000.870279
- Eberhart, R.C.; Shi, Y. *Tracking and optimizing dynamic systems with particle swarms*. In: Proceedings of the 2001 Congress on Evolutionary Computation, 2001, vol. 1, pp. 94-100. doi:10.1109/cec.2001.934376
- Eberhart, R.; Kennedy, J. *A new optimizer using particle swarm theory*. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science. IEEE, 1995, pp. 39-43. doi:10.1109/mhs.1995.494215

- Evers, G.I.; Ghalia, M.B. *Regrouping Particle Swarm Optimization: A new global optimization algorithm with improved performance consistency across benchmarks*. In: IEEE International Conference on Systems, Man and Cybernetics, 2009, pp. 3901-3908. doi:10.1109/icsmc.2009.5346625
- Fajfar, I.; Bürmen, Á.; Puhan, J. *The Nelder–Mead Simplex algorithm with perturbed centroid for high-dimensional function optimization*. Optimization Letters, 2019; vol. 13, n.5, pp. 1011-1025. doi:10.1007/s11590-018-1306-2.
- Fakhouri, H.N.; Hudaib, A.; Sleit, A. *Hybrid Particle Swarm Optimization with Sine Cosine Algorithm and Nelder–Mead Simplex for solving engineering design problems*. Arabian Journal for Science and Engineering, 2020, vol. 45, n.4, pp. 3091-3109. doi:10.1007/s13369-019-04285-9
- Fan, S.K.S.; Liang, Y.C.; Zahara, E. *Hybrid Simplex search and Particle Swarm Optimization for the global optimization of multimodal functions*. Engineering Optimization, 2004, vol. 36, n.4, pp. 401–418. doi:10.1080/0305215041000168521
- Fichtner, A. *Full Seismic Waveform Modelling and Inversion*. Springer-Verlag Berlin Heidelberg, 2011. Available from: <https://doi.org/10.1007/978-3-642-15807-0>.
- Firouzi, B.B.; Sadeghi, M.S.; Niknam, T. *A new hybrid algorithm based on PSO, SA, and k-means for cluster analysis*. International journal of innovative Computing, information and control, 2010, vol. 6, n.7, pp. 3177-3192.
- Gao, F.; Han, L. *Implementing the Nelder-Mead Simplex algorithm with adaptive parameters*. Computational Optimization and Applications, 2010, vol. 51, n.1, pp. 259-277. doi:10.1007/s10589-010-9329-3
- Gao, Y.; Song, H.; Zhang, J.; Yao, Z. *Comparison of artificial absorbing boundaries for acoustic wave equation modelling*. Exploration Geophysics, 2017, vol. 48, n.1, pp. 76-93. doi:10.1071/eg15068

- Givens, G.H.; Hoeting, J.A. *Computational statistics*. 2nd ed. Hoboken, NJ, USA: John Wiley & Sons; 2013.
- Gomes, F.I.F.N.; Bezerra, Y.S.F.; Henriques, M.V.C.; Neto, A.B.; Lucena, L.S. Estratégia híbrida de otimização sem derivadas para a inversão completa da forma da onda. HOLOS, 2017; vol. 2, pp. 2-34. <https://doi.org/10.15628/holos.2017.5445>.
- Hartigan, J.A.; Wong, M.A. *Algorithm AS 136: A K-Means clustering algorithm*. Applied Statistics, 1979; vol. 28, n.1, pp. 100-108. doi:10.2307/2346830
- Kennedy, J.; Eberhart, R. *Particle Swarm Optimization*. In: Proceedings of IEEE International Conference on Neural Networks (ICNN), 1995, vol. 4, pp. 1942-1948. Available from: <https://doi.org/10.1109/icnn.1995.488968>.
- Kennedy, J. *Small worlds and mega-minds: Effects of neighborhood topology on Particle Swarm performance*. Proceedings of IEEE Congress on Evolutionary Computation-CEC99 (Cat No 99TH8406), 1999, pp. 1391-1938. doi:10.1109/CEC.1999.785509.
- Kennedy, J. *The particle swarm: social adaptation of knowledge*. In: Proceedings of IEEE International Conference on Evolutionary Computation, 1997, pp. 303-308. Available from: <https://ieeexplore.ieee.org/document/592326>
- Kennedy, J.; Eberhart, R.; Shi, Y. *Swarm intelligence*. San Francisco: Morgan Kaufmann Publishers, 2001.
- Kirkpatrick, S.; Gelatt Jr, C.D.; Vecchi, M.P. *Optimization by Simulated Annealing*. Science. 1983, vol. 220, n. 4598, pp. 671-680. doi:10.1126/science.220.4598.671

- Koduru, P.; Das, S.; Welch, S.M. *A Particle Swarm Optimization-Nelder Mead Hybrid Algorithm for Balanced Exploration and Exploitation in Multidimensional Search Space*. In: Proceedings of International Conference on Artificial Intelligence, 2006, vol. 2, pp. 457-464.
- Lagarias, J.C.; Reeds, J.A.; Wright, M.H.; Wright, P.E. *Convergence properties of the Nelder–Mead Simplex method in low dimensions*. SIAM Journal on Optimization, 1998, vol. 9, n.1, pp. 112–147. doi:10.1137/s1052623496303470
- Lawder, J.K.; King, P.J.H., *Using state diagrams for hilbert curve mappings*. International Journal of Computer Mathematics, 2001, vol. 78, n.3, pp. 327-342. doi:10.1080/00207160108805115.
- Mazzotti, A.; Bienati, N.; Stucchi, E.; Tognarelli, A.; Aleardi, M.; Sajeva, A. *Two-grid genetic algorithm full-waveform inversion*. The Leading Edge, 2016, vol. 35, n.12, pp. 1068-1075. doi:10.1190/tle35121068.1.
- McKinnon, K.I.M. *Convergence of the Nelder-Mead Simplex method to a nonstationary point*. SIAM Journal on Optimization, 1998, vol. 9, n.1, pp. 148-158. doi:10.1137/s1052623496303482
- Mehta, V.K. *Improved Nelder–Mead algorithm in high dimensions with adaptive parameters based on Chebyshev spacing points*. Engineering Optimization, 2019, vol. 52, n.10, pp. 1814-1828. doi:10.1080/0305215X.2019.1688315.
- Medina, A.J.R.; Pulido, G.T.; Ramírez-Torres, J.G. *A comparative study of neighborhood topologies for Particle Swarm optimizers*. In Proceedings of the International Joint Conference on Computational Intelligence. SciTePress - Science and Technology Publications, 2009, vol. 1 pp. 152-159. doi:10.5220/0002324801520159; Corpus ID: 10626567.

- Metropolis, N.; Ulam, S. *The Monte Carlo method*. Journal of the American Statistical Association, 1949, vol. 44, n.247, pp. 335-341. doi:10.1080/01621459.1949.10483310
- Mitchell, M. *An introduction to Genetic Algorithms*. Cambridge, Massachusetts: MIT Press, 1996.
- Moon, B.; Jagadish, H.V.; Faloutsos, C.; Saltz, J.H. *Analysis of the clustering properties of the Hilbert space-filling curve*, IEEE Transactions on Knowledge and Data Engineering, 2001, vol. 13, n.1, pp. 124-141. Available from: <https://doi.org/10.1109/69.908985>.
- Moura, F.A.; Silva, S.A.; de Araújo, J.M.; Lucena, L.S. *Progressive matching optimisation method for FWI*. Journal of Geophysics and Engineering, 2020, vol. 17, n.2, pp.357-364. Available from: <https://doi.org/10.1093/jge/gxz121>
- Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S. *An improved grey wolf optimizer for solving engineering problems*. Expert Systems with Applications, 2021, vol. 166, 113917. <https://doi.org/10.1016/j.eswa.2020.113917>
- Nayak, J.; Kanungo, D.P.; Naik, B.; Behera, H.S. *Evolutionary Improved Swarm-Based Hybrid K-Means Algorithm for Cluster Analysis*. In: Advances in Intelligent Systems and Computing. Springer India, 2015, vol. 379, p. 343-352. Available from: https://doi.org/10.1007/978-81-322-2517-1_34.
- Nelder, J.A.; Mead, R. *A Simplex Method for Function Minimization*. The Computer Journal, 1965, vol. 7, n.4, pp. 308-313. doi:10.1093/comjnl/7.4.308.
- Olsson, D.M.; Lloyd, S.N. *The Nelder-Mead Simplex procedure for function minimization*. Technometrics, 1975, vol. 17, n.1, pp. 45-51. doi:10.1080/00401706.1975.10489269.

- Pan, J.; Wang, X.; Zhang, X.; Xu, Z.; Zhao, P.; Tian, X.; Pan, S. *2D multi-scale hybrid optimization method for geophysical inversion and its application*. Applied Geophysics, 2009, vol. 6, n.4, pp. :337-348. doi:10.1007/s11770-009-0034-x.
- Perumal, P; Dilip, M. *A hybrid approach for clustering based on combinational algorithms*, WSEAS Transactions on Computers, 2018, vol. 17, pp. 33-41.
- Price, W.L. *Global optimization by Controlled Random Search*. Journal of Optimization Theory and Applications, 1983, vol. 40, n.3, pp. 333-348. doi:10.1007/bf00933504
- Plessix, R.E. *A review of the adjoint-state method for computing the gradient of a functional with geophysical applications*. Geophysical Journal International, 2006, vol. 167, n.2, pp. 495-503. Available from: <https://doi.org/10.1111/j.1365-246X.2006.02978.x>.
- Rana, S.; Jasola, S.; Kumar, R. *A hybrid sequential approach for data clustering using K-Means and Particle Swarm Optimization algorithm*. International Journal of Engineering, Science and Technology, 2010, vol. 2, n.6, pp. 167-176.
- Ratnaweera, A.; Halgamuge S.K.; Watson H.C. *Self-organizing hierarchical Particle Swarm Optimizer with time-varying acceleration coefficients*. IEEE Transactions on Evolutionary Computation, 2004, vol. 8, pp. 240-255. doi:10.1109/TEVC.2004.826071
- Reynolds, A.C. *Boundary conditions for the numerical solution of wave propagation problems*. Geophysics, 1978, vol. 43, n.6, pp. 1099-1110. doi:10.1190/1.1440881.
- Ricker, N. *The form and nature of seismic waves and the structure of seismograms*. Geophysics, 1940, vol. 5, n.4, pp. 348-366. doi:10.1190/1.1441816.

- Riget, J.; Vesterstrøm, J.S. *A diversity-guided Particle Swarm Optimizer – The ARPSO*. Department of Computer Science, Aarhus University, Munkegade, Denmark; 2002
- Sajeve, A.; Aleardi, M.; Mazzotti, A. *Genetic algorithm full-waveform inversion: uncertainty estimation and validation of the results*. Bollettino di Geofisica Teorica ed Applicata, 2017, vol. 58, n.4, pp. 395-414. doi:10.4430/bgta0199.
- Sajeve, A.; Aleardi, M.; Stucchi, E.; Bienati, N.; Mazzotti, A. *Estimation of acoustic macro models using a genetic full-waveform inversion: Applications to the Marmousi model*. Geophysics, 2016, vol. 81, n.4, pp. R173-R184. doi:10.1190/geo2015-0198.1.
- Shi, Y.; Eberhart, R. *A modified particle swarm optimizer*. IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), 1998, pp. 69-73. Available from: <https://doi.org/10.1109/icec.1998.699146>.
- Shi, Y.; Eberhart, R.C. *Parameter selection in Particle Swarm Optimization*. In: Proceedings of the 7th International Conference on Evolutionary Programming VII, 1998, pp. 591-600. DOI 10.1007/bfb0040810
- Shi, Y.; Eberhart, R.C. *Empirical study of Particle Swarm Optimization*. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), 1999, pp. 1945-1950. doi:10.1109/cec.1999.785511.
- Steigleder, M.; McCool, M.D. *Generalized stratified sampling using the Hilbert curve*, Journal of Graphics Tools, 2003, vol. 8, n.3, pp. 41-47. doi:10.1080/10867651.2003.10487589
- Suganthan, P.N. *Particle Swarm optimiser with neighbourhood operator*. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), 1999, vol. 3, pp. 1958-1962. doi:10.1109/CEC.1999.785514.

- Surjanovic, S.; Bingham, D. *Virtual Library of Simulation Experiments: Test Functions and Datasets*. 2013 [citado em 23 de novembro de 2022]. In: Simon Fraser University site [Internet]. Burnaby: Derek Bingham 2013. Disponível em: <http://www.sfu.ca/~ssurjano/>
- Tarantola, A. *Inversion of seismic reflection data in the acoustic approximation*. Geophysics, 1984, vol. 49, n.8, pp. 1259-1266. doi:10.1190/1.1441754.
- Tarantola, A. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, 2005, Available from: <https://doi.org/10.1137/1.9780898717921>.
- Thomas, J. E. *Fundamentos de Engenharia de Petróleo*. Editora Interciência, 2ª ed., Rio de Janeiro, 2001, 271p.
- Tran, K.T.; Hiltunen, D.R. *One-Dimensional Inversion of Full Waveforms using a Genetic Algorithm*. Journal of Environmental and Engineering Geophysics, 2012, vol. 17, n.4, pp. 197-213. doi:10.2113/jeege17.4.197.
- Tran, K.T.; Hiltunen, D.R. *Two-Dimensional Inversion of Full Waveforms Using Simulated Annealing*. Journal of Geotechnical and Geoenvironmental Engineering. 2012a, vol. 138, n. 9, pp. 1075-1090. doi:10.1061/(asce)gt.1943-5606.0000685
- van Leeuwen, T.; Herrmann, F.J. *Fast waveform inversion without source-encoding*. Geophysical Prospecting, 2012b, vol. 61, n.1, pp. 10-19. doi:10.1111/j.1365-2478.2012.01096.x
- Zahara, E.; Hu, C.H. *Solving constrained optimization problems with hybrid Particle Swarm Optimization*. Engineering Optimization, 2008, vol. 40, n.11, pp. 1031-1049. doi:10.1080/03052150802265870

- Zhang, H.; Li, X.; Song, H.; Liu, S. *An adaptive subspace trust-region method for frequency-domain seismic full waveform inversion*. Computers & Geosciences, 2015, vol. 78, pp. 1-14. doi:10.1016/j.cageo.2015.02.007.
- Wang, H.; Sun, H.; Li, C.; Rahnamayan, S.; Pan, J.S. *Diversity enhanced Particle Swarm Optimization with neighborhood search*. Information Sciences, 2013, vol. 223, pp. 119-135. doi:10.1016/j.ins.2012.10.012.

Apêndice

Apêndice A

Neste capítulo são apresentados para cada função benchmark os 100 resultados obtidos pelos algoritmos ANMS, PSO classic, PSO mod e PSO-Kmeans-ANMS aplicados aos exames com 36 partículas.

Nas funções benchmark apresentadas neste texto, a notação padrão pode ser entendida como $\mathbf{X} = x_1, \dots, x_n$, onde n é o número da dimensão espacial da função.

Função Ackley

A função n-dimensional de Ackley, $f(\mathbf{X}): \mathbb{R}^n \rightarrow \mathbb{R}$, é dada por:

$$f(\mathbf{x}) = -A \exp\left(-B \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left[\frac{1}{n} \sum_{i=1}^n \cos(Cx_i)\right] + A + \exp(1),$$

onde neste trabalho foi definida para $n=2$, em que $x_1 = x$ e $x_2 = y$ incluídos no intervalo $x_i \in -32,768; 32,768$. Os valores recomendados para as variáveis são: $A=20$, $B=0,2$ e $C=2\pi$. Os valores ótimos das coordenadas e da função objetivo são $x_1=3$, $x_2=0,5$ e $f(\mathbf{X})=0$.

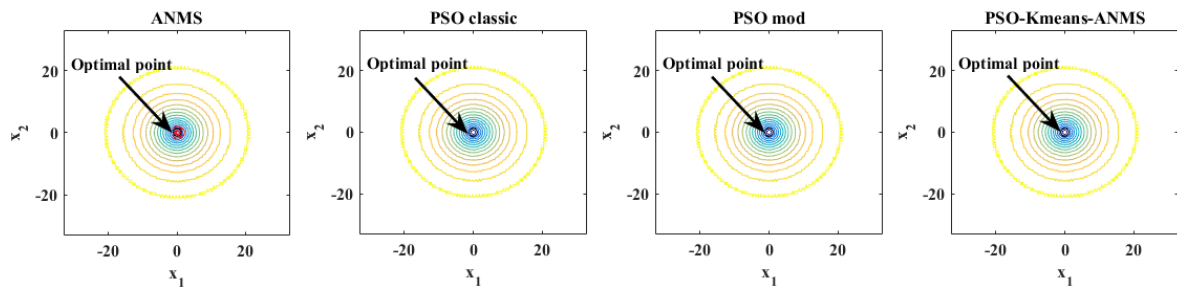


Figura 1 – Detalhes das simulações para a função Ackley.

Função Rastrigin

A função n-dimensional de Rastrigin, $f(\mathbf{X}): \Re^n \rightarrow \Re$, é dada por:

$$f(\mathbf{x}) = A \times n + \sum_{i=1}^n \left[x_i^2 - A \cos(2\pi x_i) \right]$$

onde foi definida para o caso 2D ($n=2$), em que $x_1 = x$ e $x_2 = y$ incluídos no intervalo $x_i \in -5,12; 5,12$. O valor recomendado da variável é $A=10$. Os valores ótimos das coordenadas e da função objetivo são $x_1 = 0$, $x_2 = 0$ e $f(\mathbf{X}) = 0$.

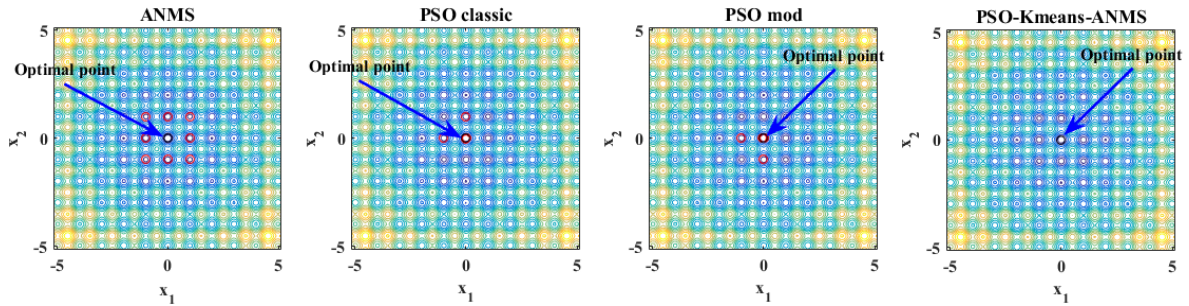


Figura 2 – Detalhes das simulações para a função Rastrigin.

Função Beale

A função n-dimensional de Beale, $f(\mathbf{X}): \Re^n \rightarrow \Re$, é dada por:

$$f(\mathbf{x}) = (1,5 - x_1 + x_1 x_2)^2 + (2,25 - x_1 + x_1 x_2^2)^2 + (2,625 - x_1 + x_1 x_2^3)^2$$

onde neste trabalho foi definida no espaço bidimensional, ou seja, $n=2$ tal que $x_1 = x$ e $x_2 = y$ ambos incluídos no intervalo $x_i \in -4,5; 4,5$. Os valores ótimos das coordenadas e da função objetivo são $x_1 = 3$, $x_2 = 0,5$ e $f(\mathbf{X}) = 0$.

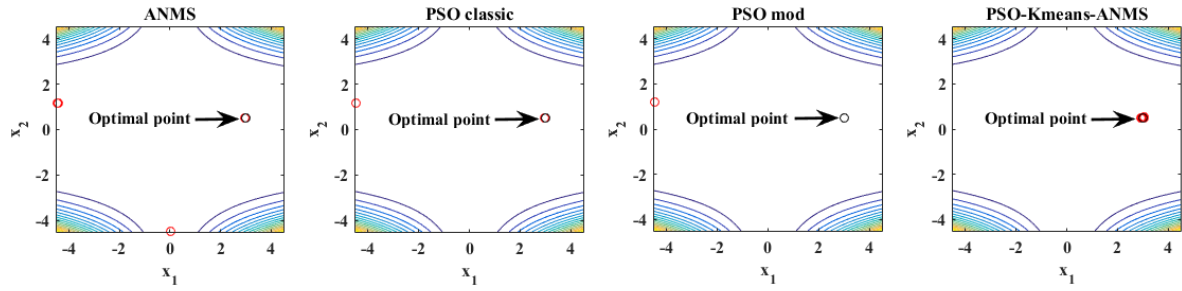


Figura 3 – Detalhes das simulações para a função Beale.

Função Rosenbrock

A função n-dimensional de Rosenbrock, $f(\mathbf{X}): \mathbb{R}^n \rightarrow \mathbb{R}$, é dada por:

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} 100 \left[(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$$

onde neste trabalho foi definido $n=2$, ou seja, duas dimensões em que $x_1 = x$ e $x_2 = y$ incluídos no intervalo $x_i \in -10; 10$. Os valores ótimos das coordenadas e da função objetivo são $x_1 = 1$, $x_2 = 1$ e $f(\mathbf{X}) = 0$.

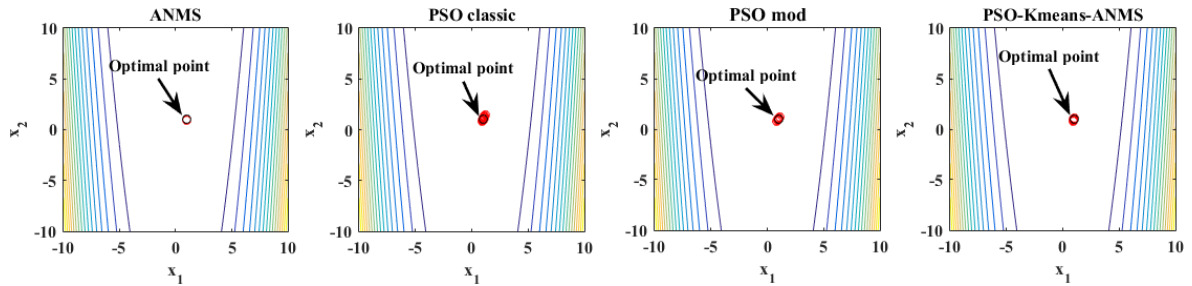


Figura 4 – Detalhes das simulações para a função Rosenbrock.

Função Sphere

A função n-dimensional de Sphere, $f(\mathbf{X}): \mathbb{R}^n \rightarrow \mathbb{R}$, é dada por:

$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$$

onde neste trabalho foi definido $n=2$, ou seja, duas dimensões em que $x_1 = x$ e $x_2 = y$ incluídos no intervalo $x_i \in -5,12; 5,12$. Os valores ótimos das coordenadas e da função objetivo são $x_1 = 0$, $x_2 = 0$ e $f(\mathbf{X}) = 0$.

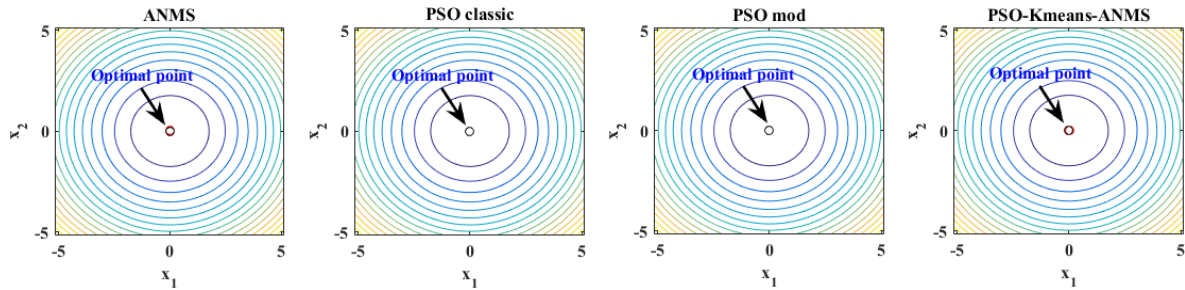


Figura 5 – Detalhes das simulações para a função Sphere.

Função Zakharov

A função n-dimensional de Zakharov, $f(\mathbf{X}): \mathbb{R}^n \rightarrow \mathbb{R}$, é dada por:

$$f(\mathbf{x}) = \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n 0,5ix_i \right)^2 + \left(\sum_{i=1}^n 0,5ix_i \right)^4$$

onde neste trabalho foi definida para $n=2$, ou seja, duas dimensões em que $x_1 = x$ e $x_2 = y$ incluídos no intervalo $x_i \in -10; 10$. Os valores ótimos das coordenadas e da função objetivo são $x_1 = 0$, $x_2 = 0$ e $f(\mathbf{X}) = 0$.

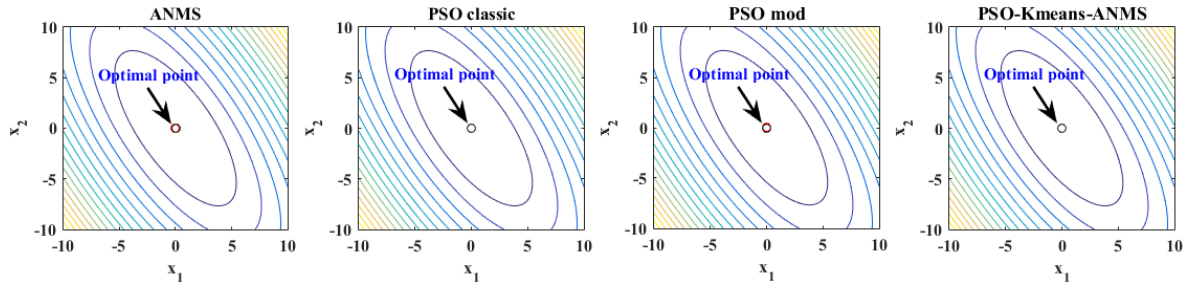


Figura 6 – Detalhes das simulações para a função Zakharov.

Função Michalewicz

A função n-dimensional de Michalewicz, $f(\mathbf{X}) : \mathcal{R}^n \rightarrow \mathcal{R}$, é dada por:

$$f(\mathbf{X}) = -\sum_{i=1}^n \sin(x_i) \sin^{2m} \left(\frac{ix_i^2}{\pi} \right)$$

onde neste trabalho foi definida para $n=2$, ou seja, duas dimensões tal que $x_1 = x$ e $x_2 = y$ ambos incluídos no intervalo $x_i \in 0; \pi$. Os valores ótimos das coordenadas e da função objetivo são $x_1 = 2,2$, $x_2 = 1,57$ e $f(\mathbf{X}) = -1,8013$.

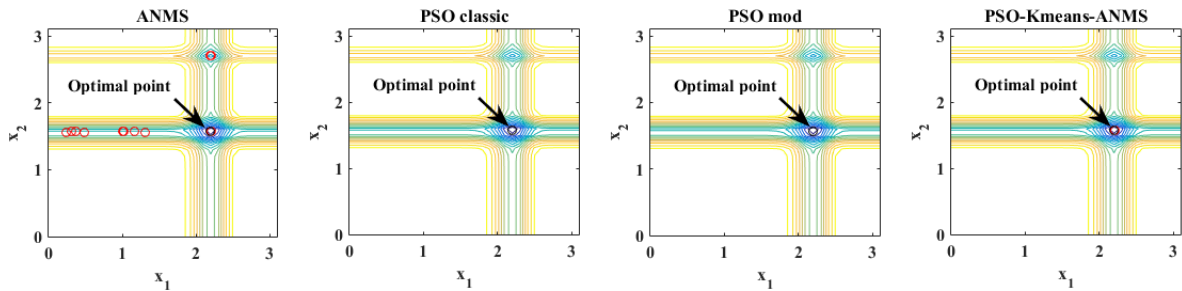


Figura 7 – Detalhes das simulações para a função Michalewicz.

Função Styblinski-Tang

A função n-dimensional de Styblinski-Tang, $f(\mathbf{X}): \mathbb{R}^n \rightarrow \mathbb{R}$, é dada por:

$$f(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$$

onde neste trabalho foi definida para o caso 2D, ou seja, duas dimensões em que $x_1 = x$ e $x_2 = y$ incluídos no intervalo $x_i \in -5; 5$. Os valores ótimos das coordenadas e da função objetivo são $x_1 = -2,903534$, $x_2 = -2,903534$ e $f(\mathbf{X}) = -78,3320$.

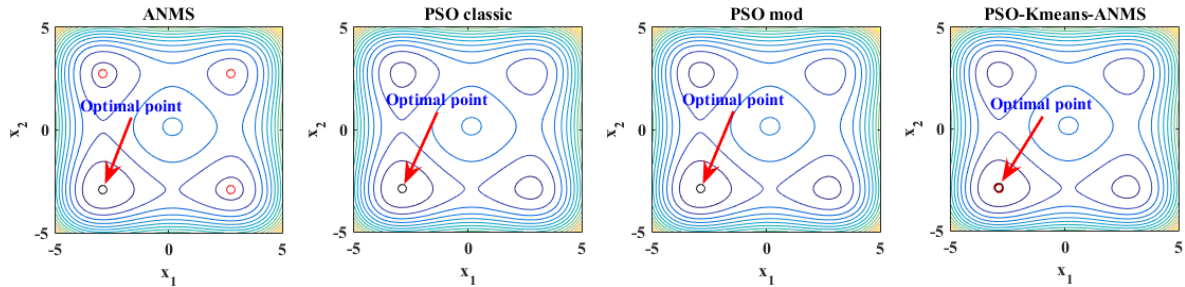


Figura 8 – Detalhes das simulações para a função Styblinski-Tang.

Função F12

A função n-dimensional de F12, $f(\mathbf{X}): \mathbb{R}^n \rightarrow \mathbb{R}$, é dada por:

$$f(\mathbf{x}) = \frac{\pi}{n} \left[10 \sin(\pi g_1) + \sum_{i=1}^{n-1} (g_i - 1)^2 [1 + 10 \sin^2(\pi g_{i+1})] + (g_n - 1)^2 \right] + \sum_{i=1}^n u(x_i, 10, 100, 4)$$

$$g_i = 1 + \frac{x_i + 1}{4},$$

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

onde foi definida para o caso 2D, ou seja, $n=2$ em que $x_1 = x$ e $x_2 = y$ incluídos no intervalo $x_i \in -10; 10$. Os valores ótimos das coordenadas e da função objetivo são $x_1 = -1$, $x_2 = -1$ e $f(\mathbf{X}) = 0$.

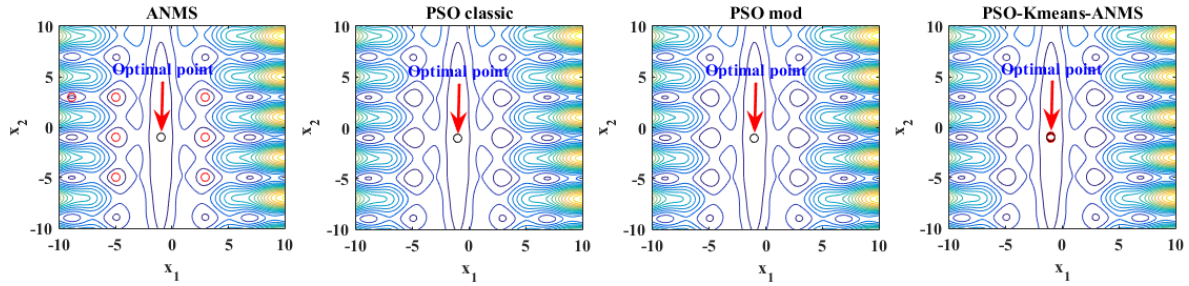


Figura 9 – Detalhes das simulações para a função F12.

Função F22

A função n-dimensional de F22, $f(\mathbf{X}): \mathbb{R}^n \rightarrow \mathbb{R}$, é dada por:

$$f(\mathbf{X}) = -\sum_{j=1}^7 \left[(x_i - a_{j,i})(x_i - a_{j,i})^T + c_j \right]^{-1}, \quad i = 1, \dots, n$$

com

$$\mathbf{A} = [a_{j,i}]_{4 \times 10} = \begin{bmatrix} 4 & 1 & 8 & 6 & 3 & 2 & 5 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 5 & 1 & 2 & 3,6 \\ 4 & 1 & 8 & 6 & 3 & 2 & 3 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 3 & 1 & 2 & 3,6 \end{bmatrix}$$

e $c_j = [0,1; 0,2; 0,2; 0,4; 0,4; 0,6; 0,3; 0,7; 0,5; 0,5]$, onde foi definida para $n=2$, ou seja, duas dimensões em que $x_1 = x$ e $x_2 = y$ incluídos no intervalo $x_i \in -5; 5$. Os valores ótimos das coordenadas e da função objetivo são $x_1 = 1,02$, $x_2 = 1,02$ e $f(\mathbf{X}) = -0,524$.

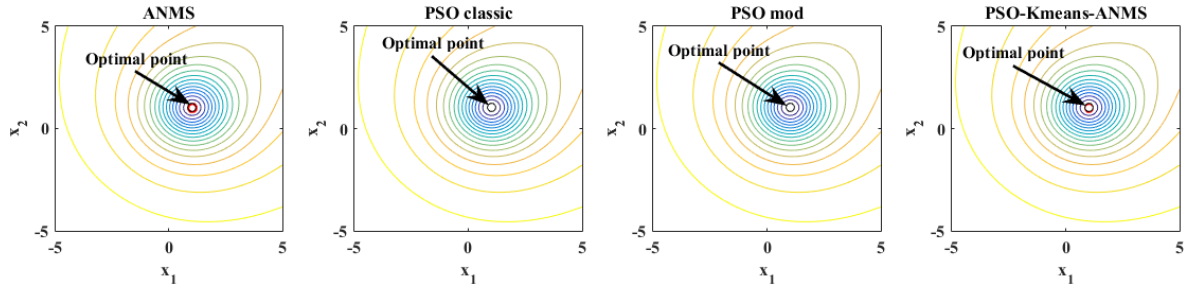


Figura 10 – Detalhes das simulações para a função F22.

Função F2

A função n-dimensional de F2, $f(\mathbf{X}): \mathbb{R}^n \rightarrow \mathbb{R}$, é dada por:

$$f(\mathbf{x}) = \sum_{j=1}^n |x_j| + \prod_{i=1}^n |x_i|$$

onde neste trabalho foi definida para $n=2$, em que $x_1 = x$ e $x_2 = y$ incluídos no intervalo $x_i \in -10; 10$. Os valores ótimos das coordenadas e da função objetivo são $x_1 = 0$, $x_2 = 0$ e $f(\mathbf{X}) = 0$.

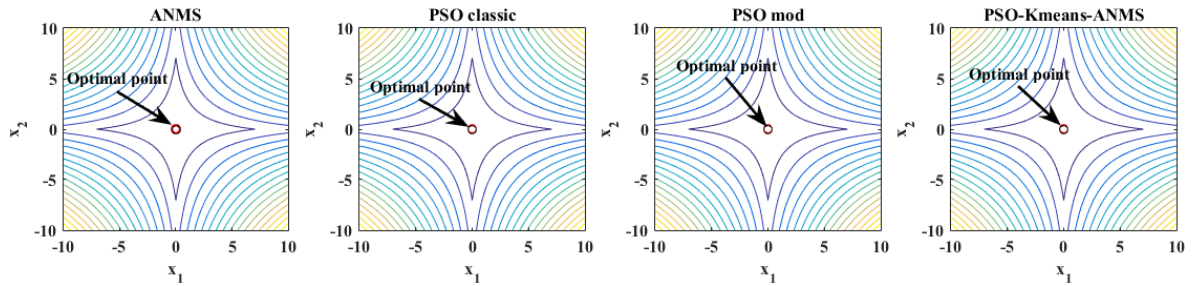


Figura 11 – Detalhes das simulações para a função F2.

Função Peaks

A função n-dimensional de Peaks, $f(\mathbf{X}): \mathbb{R}^n \rightarrow \mathbb{R}$, é dada por:

$$f(\mathbf{x}) = 3(1-x_1)^2 e^{[-x_1^2-(x_2+1)^2]} - 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right) e^{(-x_1^2-x_2^2)} - \frac{1}{3} e^{[-(x_1+1)^2-x_2^2]}$$

onde foi definida para $n=2$, ou seja, duas dimensões em que $x_1 = x$ e $x_2 = y$ incluídos no intervalo $x_i \in -4; 4$. Os valores ótimos das coordenadas e da função objetivo são $x_1 = 0,2283$, $x_2 = -1,6255$ e $f(\mathbf{X}) = -6,5511$.

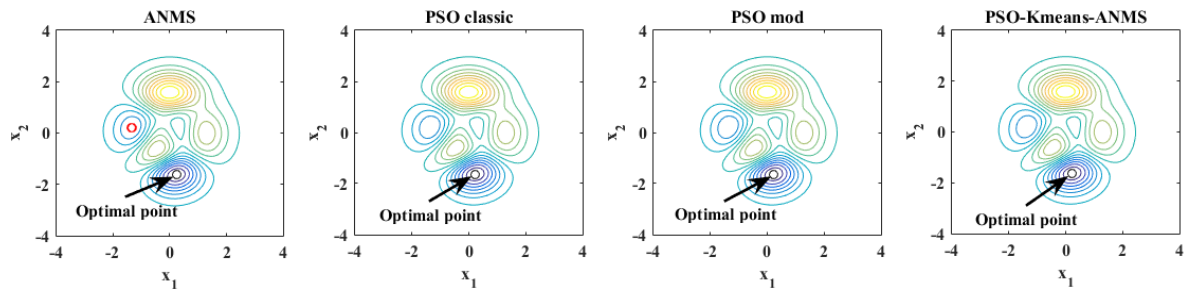


Figura 12 – Detalhes das simulações para a função Peaks.

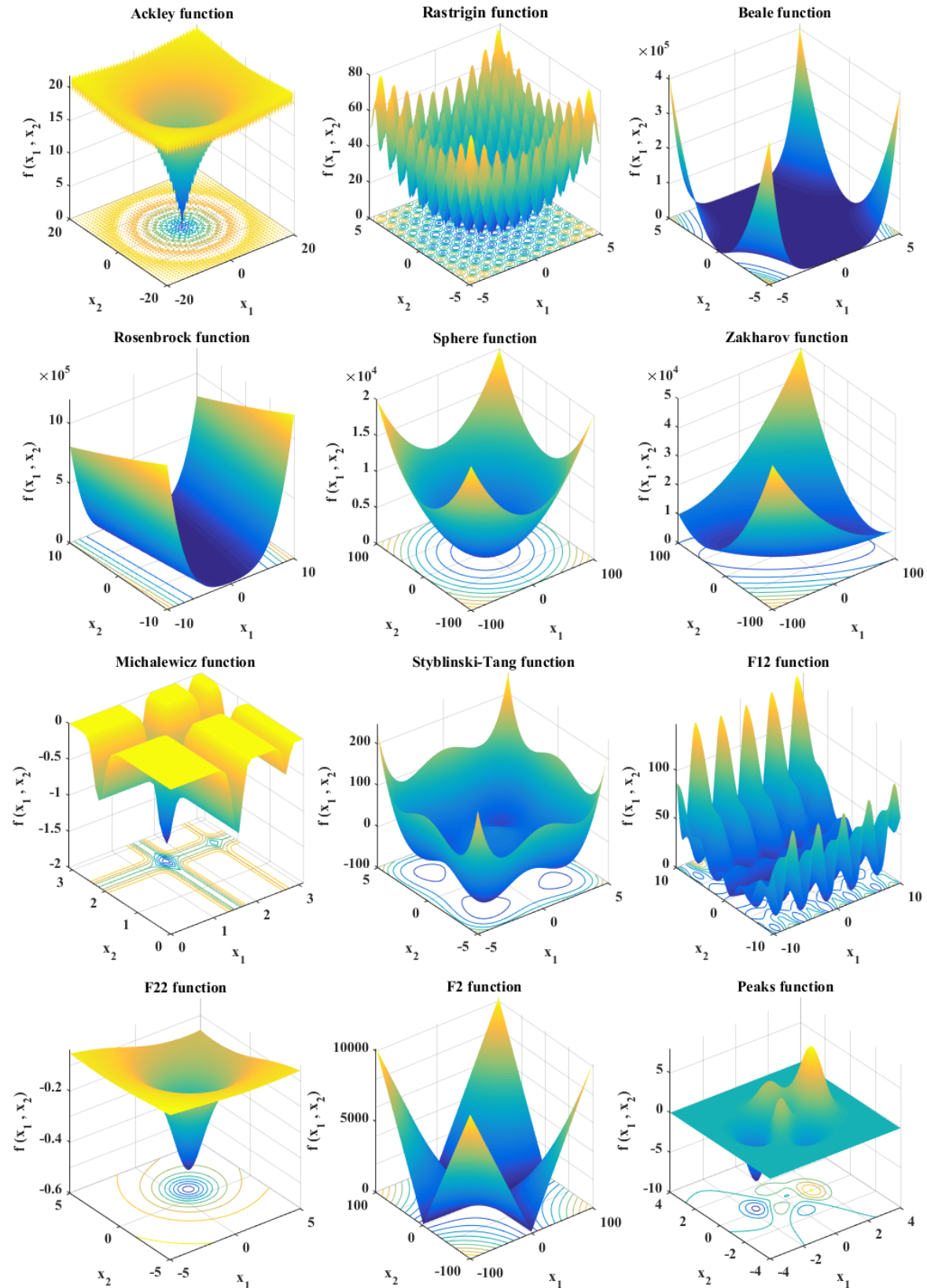


Figura 13 – Funções Benchmark para validação do algoritmo híbrido.

Apêndice B

Publicações Vinculadas

Seguem relacionados abaixo os trabalhos publicados ou submetidos em periódicos nacionais e internacionais, apresentados em eventos internacionais e demais textos do autor e de seus colaboradores que tenham conexão com a tese aqui apresentada.

- Apresentação oral em evento internacional:

Aguiar, R.; Neto, A.B.; Bezerra, Y.S.F.; Henriques, M.V.C.; Nascimento, H.A.D.; Lucena, L.S. *A Global Optimization DFO-CRS Strategy for 1D Full Waveform Inversion*. 4th Conference of Computational Interdisciplinary Sciences, held in the Instituto Nacional de Pesquisas Espaciais (INPE), Brazil, 2016.

- Artigo complemento submetido em periódico nacional:

Bezerra, Y.S.F.; Neto, A.B.; Aguiar, R.; Nascimento, H.A.D. Algoritmo de Otimização Global com Busca Aleatória Controlada Aplicado à Inversão da Forma Completa da Onda 1D. *Holos*, 2023.

- Artigo completo publicado em periódico internacional:

Aguiar, R.; Neto, A.B.; Bezerra, Y.S.F.; Henriques, M.V.C.; Nascimento, H.A.D.; Lucena, L.S. *A Global Optimization DFO-CRS Strategy for 1D Full Waveform Inversion*. *Journal of Computational Interdisciplinary Sciences*, 2019, vol. 10, n.1, pp. 21-30. doi: 10.6062/jcis.2019.10.01.0156

Aguiar, R.; Neto, A.B.; Bezerra, Y.S.F.; Nascimento, H.A.D.; Lucena, L.S.; de Freitas, J.E. *A new hybrid optimization approach using PSO, Nelder-Mead Simplex and Kmeans clustering algorithms for 1D Full Waveform Inversion*. *Plos One*, 2022, vol. 17, n.12, pp. e0277900. In S. Mirjalili (Ed.). *Public Library of Science (PLoS)*. doi: 10.1371/journal.pone.0277900