

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO ASSOCIADO, EM ENGENHARIA
ELÉTRICA

GISELE ALVES SANTANA

**HALTON-CHAOS- β -PSO: UMA NOVA ABORDAGEM
METAHEURÍSTICA DE OTIMIZAÇÃO POR ENXAME DE
PARTÍCULAS APLICADA À PROBLEMAS DE ENGENHARIA
ELÉTRICA**

TESE

CORNÉLIO PROCÓPIO

2021

GISELE ALVES SANTANA

**HALTON-CHAOS- β -PSO: UMA NOVA ABORDAGEM
METAHEURÍSTICA DE OTIMIZAÇÃO POR ENXAME DE
PARTÍCULAS APLICADA À PROBLEMAS DE ENGENHARIA
ELÉTRICA**

**Halton-Chaos- β -PSO: a new metaheuristic approach
based on particle swarm optimization applied to
electrical engineering problems**

Tese apresentada ao Programa de Pós-Graduação
Associado, em Engenharia Elétrica da Universidade
Tecnológica Federal do Paraná como requisito par-
cial para obtenção do grau de “Doutor em Enge-
nharia Elétrica” – Área de Concentração: Sistemas
Eletrônicos.

Orientador: Prof. Dr. Taufik Abrão

CORNÉLIO PROCÓPIO

2021



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Esta licença permite download e compartilhamento do trabalho desde que sejam atribuídos créditos ao(s) autor(es), sem a possibilidade de alterá-lo ou utilizá-lo para fins comerciais. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



**Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio**



GISELE ALVES SANTANA

HALTON-CHAOS-BETA-PSO: UMA NOVA ABORDAGEM METAHEURÍSTICA BASEADA EM ENXAME DE PARTÍCULAS APLICADA À PROBLEMAS DE ENGENHARIA ELÉTRICA.

Trabalho de pesquisa de doutorado apresentado como requisito para obtenção do título de Doutora Em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Sistemas Eletrônicos.

Data de aprovação: 30 de Julho de 2021

Prof Taufik Abrao, Doutorado - Universidade Estadual de Londrina (Uel)

Prof Andre Luis Machado Martinez, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Cristiano Magalhaes Panazio, Doutorado - Universidade de São Paulo (Usp)

Prof Jose Carlos Pizolato Junior, Doutorado - Universidade Federal de São Carlos (Ufscar)

Prof Marcelo Favoretto Castoldi, Doutorado - Universidade Tecnológica Federal do Paraná

Prof.a Marcia Marcondes Altimari Samed, Doutorado - Universidade Estadual de Maringá (Uem)

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 30/07/2021.

AGRADECIMENTOS

Esta fase da minha vida foi muito especial e gostaria de agradecer primeiramente a Deus pela força para alcançar minha meta e saúde, principalmente nesta fase difícil com a pandemia afetando o mundo todo, e inclusive, vitimizando alguns de meus familiares.

Agradeço também à minha família, principalmente à minha mãe, que sempre esteve presente em todos os passos da minha vida acadêmica. Agradeço especialmente ao meu marido Rubinho, que teve muita compreensão e ofereceu grande apoio durante esta fase.

Ao Prof. Dr. Fábio Renan Durand (*in memoriam*), que foi meu orientador por dois anos, agradeço pela confiança em mim depositada. Agradeço pela oportunidade, não só por sua orientação nos anos iniciais do curso, mas também pelo aprendizado com sua sabedoria, competência, determinação, força e coragem.

Ao Prof. Dr. Taufik Abrão, pelo apoio e disponibilidade em prosseguir com a orientação deste trabalho, pelo qual tenho muita admiração. Foi um grande aprendizado trabalhar ao seu lado.

Aos meus colegas de estudo e aos professores do programa, pela confiança, incentivo e apoio em todos os momentos e, especialmente, a colega Cristiane, por seu apoio e amizade.

“Que os nossos esforços estejam sempre focados no desafio às impossibilidades. Todas as grandes conquistas humanas vieram daquilo que parecia impossível.”

(Charles Chaplin)

RESUMO

SANTANA, Gisele Alves. HALTON-CHAOS- β -PSO: UMA NOVA ABORDAGEM METAHEURÍSTICA DE OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS APLICADA À PROBLEMAS DE ENGENHARIA ELÉTRICA. 140 f. Tese – Programa de Pós-Graduação Associado, em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2021.

Os problemas do mundo real são complexos, multidimensionais e multimodais, o que incentiva os cientistas a desenvolverem métodos mais eficazes e eficientes para a solução destes problemas. As metaheurísticas tentam emular biologicamente a natureza evolutiva e adaptativa dos seres vivos por meio de uma série de técnicas e têm mostrado melhor desempenho que os métodos tradicionais. Este trabalho propõe uma nova abordagem metaheurística, chamada Halton-Chaos-Beta-PSO, que é baseada no algoritmo tradicional de Otimização por Enxame de Partículas. O algoritmo Halton-Chaos-Beta-PSO utiliza a Distribuição Beta para geração de variáveis aleatórias com o objetivo de aumentar a diversidade e auxiliar na exploração de regiões encobertas no espaço de busca durante o procedimento de otimização. Além disso, um fator de amortecimento baseado em mapas caóticos é utilizado para melhorar o processo de atualização do valor global, visando evitar a convergência prematura e aumentar a diversificação sem perda na capacidade de exploração do algoritmo. Além disso, a sequência de Halton é implantada para melhorar a qualidade da geração da população inicial, visando proporcionar a cobertura do espaço viável de busca de maneira adequada. A metaheurística proposta é testada e validada em um conjunto de funções Benchmark e seu desempenho é comparado com o algoritmo PSO tradicional em termos de velocidade de convergência, robustez, erro quadrático médio normalizado e número de FLOPs. Para todas as metaheurísticas comparadas, o ajuste ótimo dos parâmetros é investigado a fim de melhorar a qualidade das soluções. Para investigar a aplicação em problemas de Engenharia Elétrica, a abordagem proposta é utilizada para resolver o problema de alocação de potência em redes PON-OCDMA e para a estimação de parâmetros em um motor de indução trifásico (MIT). Para o problema de alocação de potência, o algoritmo demonstrou compromisso promissor entre desempenho-complexidade, especialmente para altas dimensionalidades do problema, assim como uma melhoria significativa da eficiência energética do sistema óptico quando o procedimento de alocação de taxa de potência é implantado dinamicamente em aplicações em tempo real. Em relação ao problema de estimação de parâmetros do MIT, a metaheurística proposta conseguiu estimar os parâmetros com maior rapidez, obtendo maior qualidade nas soluções, mesmo em cenários de mudança de parâmetros durante a operação do motor.

Palavras-chave: Metaheurística, Otimização por Enxame de Partículas, Controle de Potência, Redes Ópticas, Estimação de Parâmetros, Motor de Indução Trifásico, Funções Benchmark

ABSTRACT

SANTANA, Gisele Alves. HALTON-CHAOS- β -PSO: A NEW OPTIMIZATION METAHEURISTIC APPROACH APPLIED IN ELECTRICAL ENGINEERING PROBLEMS . 140 f. Tese – Programa de Pós-Graduação Associado, em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2021.

Real-world problems are complex as they are multidimensional and multimodal in nature that encourages scientists to develop better and efficient problem-solving methods. Metaheuristic algorithms attempt to biologically emulate the adaptive evolutionary nature of living beings via a series of techniques and have shown better performances than the traditional methods. This work proposes a new metaheuristic approach, namely Halton-Chaos-Beta-PSO, based on traditional Particle Swarm Optimization algorithm. The Halton-Chaos-Beta-PSO algorithm uses Beta Distribution for random variables generation aiming to increase the diversity while aid the exploration (diversification) of undercover regions in the search space during the optimization procedure. Besides, a damping factor based on chaotic maps is used to the best global value updating, aiming to limit the dominance of the best global particle value, avoid premature convergence and increase diversification without loss in the exploitation capability of the algorithm. In addition, the Halton sequence is deployed to improve the quality of the initial population generation in the first step to cover the feasible space properly. The proposed metaheuristic is tested and validated on a set of benchmark test functions and its performance is compared with the traditional PSO algorithm in terms of convergence velocity, normalized mean squared error and number of FLOPs. For all compared metaheuristics, the optimum tuning of parameters is investigated in order to improve the quality of the solutions. To investigate its application in Electrical Engineering problems, the proposed approach is used to solve the problem of power allocation in PON-OCDMA networks and for the estimation of parameters in a three-phase induction motor (TIM). For the power allocation problem, the algorithm demonstrates a promising performance-complexity trade-off, especially for high dimensionality of problems and demonstrates an improvement in the energy efficiency of the optical system when the power rate allocation procedure must be dynamically implemented in real time applications. Regarding the TIM parameter estimation problem, the Halton-Chaos-Beta-PSO algorithm is able to estimate the parameters more quickly, obtaining higher quality in the solutions, even in scenarios of changing parameter during the operation of the motor.

Keywords: Metaheuristic, Particle Swarm Optimization, Power Control, Optical Networks, Parameter Estimation, Three-phase Induction Motor, Benchmark Functions

LISTA DE FIGURAS

FIGURA 1	– Pontos máximo e mínimo de uma função.	30
FIGURA 2	– Classificação dos Métodos Heurísticos.	31
FIGURA 3	– Distribuição Beta com vários valores de p e q	34
FIGURA 4	– Tipos de Mapas Caóticos.	35
FIGURA 5	– 500 partículas iniciais para o PSO com (a) Sequência Randômica (b) Sequência de Halton.	37
FIGURA 6	– 1000 pontos gerados por (a) Sequência de Halton Embaralhada; (b) Sequência de Halton.	38
FIGURA 7	– Mínimo Global da Função Ackley de $d = 2$ dimensões.	41
FIGURA 8	– Função de teste Ackley.	42
FIGURA 9	– Função de Teste Griewank.	43
FIGURA 10	– Função Griewank ampliada.	43
FIGURA 11	– Mínimos Globais da Função Griewank.	44
FIGURA 12	– Função de teste Rosenbrock e respectivo mínimo global.	45
FIGURA 13	– Mínimo Global da Função Rastrigin.	46
FIGURA 14	– Função de teste Rastrigin.	47
FIGURA 15	– Deslocamento das partículas no algoritmo PSO Tradicional.	49
FIGURA 16	– Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Ackley considerando diferentes valores para o parâmetro p	59
FIGURA 17	– Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Griewank considerando diferentes valores para o parâmetro p	60
FIGURA 18	– Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Rastrigin considerando diferentes valores para o parâmetro p	61
FIGURA 19	– Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Rosenbrock considerando diferentes valores para o parâmetro p	62
FIGURA 20	– Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Ackley considerando diferentes valores para o parâmetro q	64
FIGURA 21	– Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Griewank considerando diferentes valores para o parâmetro q	65
FIGURA 22	– Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Rastrigin considerando diferentes valores para o parâmetro q	66
FIGURA 23	– Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Rosenbrock considerando diferentes valores para o parâmetro q	67
FIGURA 24	– Número de iterações necessárias para a convergência da metaheurística	

	C-PSO com a função Ackley considerando diferentes valores para o fator de amortecimento ξ	69
FIGURA 25	– Número de iterações necessárias para a convergência da metaheurística C-PSO com a função Griewank considerando diferentes valores para o fator de amortecimento ξ	70
FIGURA 26	– Número de iterações necessárias para a convergência da metaheurística C-PSO com a função Rastringin considerando diferentes valores para o fator de amortecimento ξ	71
FIGURA 27	– Número de iterações necessárias para a convergência da metaheurística C-PSO com a função Rosenbrock considerando diferentes valores para o fator de amortecimento ξ	72
FIGURA 28	– Número de iterações necessárias para a convergência da metaheurística H-PSO com a função Ackley considerando diferentes valores para a base do número primo b	74
FIGURA 29	– Número de iterações necessárias para a convergência da metaheurística H-PSO com a função Griewank considerando diferentes valores para a base do número primo b	75
FIGURA 30	– Número de iterações necessárias para a convergência da metaheurística H-PSO com a função Rastringin considerando diferentes valores para a base do número primo b	76
FIGURA 31	– Número de iterações necessárias para a convergência da metaheurística H-PSO com a função Rosenbrock considerando diferentes valores para a base do número primo b	77
FIGURA 32	– Número de iterações necessárias para a convergência das metaheurísticas com a função Ackley.	79
FIGURA 33	– Número de iterações necessárias para a convergência das metaheurísticas com a função Griewank.	81
FIGURA 34	– Número de iterações necessárias para a convergência das metaheurísticas com a função Rastringin.	84
FIGURA 35	– Número de iterações necessárias para a convergência das metaheurísticas com a função Rosenbrock.	87
FIGURA 36	– Número de FLOPs necessários para a convergência das metaheurísticas com a função Ackley considerando diferentes dimensões d	89
FIGURA 37	– Número de FLOPs necessários para a convergência das metaheurísticas com a função Griewank considerando diferentes dimensões.	90
FIGURA 38	– Número de FLOPs necessários para a convergência das metaheurísticas com a função Rastringin considerando diferentes dimensões.	90
FIGURA 39	– Número de FLOPs necessários para a convergência das metaheurísticas com a função Rosenbrock considerando diferentes dimensões.	91
FIGURA 40	– Número de iterações necessárias para a convergência das metaheurísticas com a função Ackley.	92
FIGURA 41	– Número de iterações necessárias para a convergência das metaheurísticas com a função Griewank.	93
FIGURA 42	– Número de iterações necessárias para a convergência das metaheurísticas com a função Rastringin.	94
FIGURA 43	– Número de iterações necessárias para a convergência das metaheurísticas com a função Rosenbrock.	95
FIGURA 44	– Circuito Equivalente do MIT na referência $dq0$	99

FIGURA 45	– Processo de Estimação de Parâmetros.	101
FIGURA 46	– Número de iterações necessárias para a convergência do algoritmo β -PSO considerando diferentes valores para os parâmetros a) p e b) q	106
FIGURA 47	– Número de iterações necessárias para a convergência do algoritmo C-PSO considerando diferentes valores de fator de amortecimento.	107
FIGURA 48	– Número de iterações necessárias para a convergência do algoritmo H-PSO considerando diferentes valores de número primos para a base b	108
FIGURA 49	– Média de 50 realizações dos algoritmos para a estimação de parâmetros do MIT: a) Resistência do estator; b) Indutância do estator; c) Momento de Inércia.	111
FIGURA 50	– Posição final do enxame: a) PSO; b) BetaPSO; c) ChaosPSO; d) HaltonChaosBetaPSO	112
FIGURA 51	– MSE para todos os algoritmos de estimação de parâmetros com média de 50 realizações.	113
FIGURA 52	– Número de iterações necessárias para a convergência do algoritmo β -PSO considerando diferentes valores para o parâmetro: (a) p ; (b) q	121
FIGURA 53	– Número de iterações necessárias para a convergência do algoritmo H-PSO considerando diferentes valores da base b	122
FIGURA 54	– Soma de potência transmitida vs. número de iterações para o PSO, β -PSO, Chaos-PSO and Halton-Chaos- β -PSO, para diferentes números de nós óticos (usuários).	124
FIGURA 55	– Convergência versus o número de iterações para os algoritmos PSO, β -PSO, Chaos-PSO and Halton-Chaos- β -PSO, para diferentes números de nós óticos (usuários).	125
FIGURA 56	– NMSE versus número de iterações para os algoritmos PSO, β -PSO, Chaos-PSO and Halton-Chaos- β -PSO para diferentes números de nós óticos (usuários).	127
FIGURA 57	– Sum-rate vs the número de iterações para os algoritmos PSO, ALPSO, β -PSO, C-PSO and H-C- β -PSO para diferentes números de nós óticos.	128
FIGURA 58	– Taxa individual versus número de iterações para o algoritmo Halton-Chaos- β -PSO com diferentes números de nós ótico (usuários).	129
FIGURA 59	– FLOPs consumidos com o aumento do número de nós para os algoritmos de alocação de potência.	130

LISTA DE TABELAS

TABELA 1	– Permutações dos 16 primeiros números primos obtidos pelo método B&W.	38
TABELA 2	– Valores dos domínios das Funções Benchmark.	56
TABELA 3	– Parâmetros da Metaheurística PSO Tradicional para a função Ackley. .	57
TABELA 4	– Parâmetros da Metaheurística PSO Tradicional para a função Griewank. .	57
TABELA 5	– Parâmetros da Metaheurística PSO Tradicional para a função Rastrigin.	57
TABELA 6	– Parâmetros da Metaheurística PSO Tradicional para a função Rosenbrock.	57
TABELA 7	– Parâmetro p da Metaheurística β -PSO para as funções Benchmark. . .	63
TABELA 8	– Parâmetro q da Metaheurística β -PSO para as funções Benchmarks. . .	68
TABELA 9	– Parâmetro p da Metaheurística C-PSO para as funções Benchmarks. .	68
TABELA 10	– Parâmetro p da Metaheurística H-PSO para as funções Benchmarks. .	73
TABELA 11	– Resultados numéricos das Metaheurísticas para a função Ackley.	80
TABELA 12	– Resultados numéricos das Metaheurísticas para a função Griewank. ...	82
TABELA 13	– Resultados numéricos das Metaheurísticas para a função Rastrigin. . .	85
TABELA 14	– Resultados numéricos das Metaheurísticas para a função Rosenbrock. .	88
TABELA 15	– Número de FLOPs para a função Ackley considerando o aumento das dimensões.	88
TABELA 16	– Número de FLOPs para a função Griewank considerando o aumento das dimensões.	88
TABELA 17	– Número de FLOPs para a função Rastrigin considerando o aumento das dimensões.	89
TABELA 18	– Número de FLOPs para a função Rosenbrock considerando o aumento das dimensões.	89
TABELA 19	– Robustez das metaheurísticas PSO, β -PSO, C-PSO e H-C- β -PSO para as funções Benchmarks.	96
TABELA 20	– Parâmetros do MIT.	105
TABELA 21	– Valores dos Parâmetros do PSO	105
TABELA 22	– Número de FLOPs para o problema de estimação de parâmetros do MIT, obtidos com o programa COUNT FLOPS (QIAN, 2015).	113
TABELA 23	– Valores dos Parâmetros da Rede PON-OCDMA Network	118
TABELA 24	– Classes de tráfego e parâmetros de código	119
TABELA 25	– Variáveis de consumo de potência	120
TABELA 26	– Valores dos Parâmetros do PSO	120
TABELA 27	– Número de FLOPs psara o problema de alocação de taxa de potência com o aumento do número de nós óticos.	126
TABELA 28	– Número de iterações necessárias para a convergência dos algoritmos considerando o amento de nós ópticos	131

LISTA DE SÍMBOLOS

P	Tamanho da população do enxame
S	Espaço de busca
p	Parâmetro do formato da distribuição Beta
q	Parâmetro do formato da distribuição Beta
Π_p	Operador de permutação para os possíveis valores de $b_1(i)$
σ	Subconjuntos de números primos
d	Dimensão do espaço de busca
R	Robustez algorítmica
Sc	Sucesso de convergência
c_1	Coeficiente de aprendizado individual
c_2	Coeficiente de aprendizado global
ζ	Tolerância de convergência dos métodos analisados
g_i^{B}	Melhor posição assumida pelo enxame até a iteração i nas meta-heurísticas
B_1	Matriz diagonal, na qual seus elementos são valores aleatórios geradas pela distribuição Beta
B_2	Matriz diagonal, na qual seus elementos são valores aleatórios geradas pela distribuição Beta
e	Fator de amortecimento dos mapas caóticos
L	Mapa caótico logístico
V_{sd}	Tensão do estator no eixo d
V_{sq}	Tensão do estator no eixo q
V_{rd}	Tensão do rotor no eixo d
V_{rq}	Tensão do rotor no eixo q
i_{sd}	Corrente do estator no eixo d
i_{sq}	Corrente do estator no eixo q
i_{rd}	Corrente do rotor no eixo d
i_{rq}	Corrente do rotor no eixo q
λ_{sq}	Fluxo magnético do estator no eixo q
λ_{sd}	Fluxo magnético do estator no eixo d
λ_{rq}	Fluxo magnético do rotor no eixo q
λ_{rd}	Fluxo magnético do rotor no eixo d
R_s	Resistência do estator
R_r	Resistência do rotor
L_r	Indutância do rotor
L_s	Indutância do estator
ω_d	Velocidade angular do motor
J	Momento de inércia do motor
p_{plant}	Parâmetros a serem identificados no motor
f	Frequência do motor
V	Tensão do motor
BER_i	Taxa de erro de bit tolerável para o i -ésimo usuário

γ_i	SNIR do i -ésimo usuário
γ_i^*	SINR alvo do i -ésimo usuário
Γ_i	CINR do i -ésimo usuário
Γ^*	CINR alvo do i -ésimo usuário
$\Gamma^+(\cdot)$	Função obtida transformando a restrição de desigualdade da CINR em uma restrição de igualdade
$E[\cdot]$	Operador esperança
ε_k	Parâmetro de tolerância do k -ésimo subproblema do ALM
F	Ganho de processamento
F_i	Ganho de processamento do i -ésimo usuário
F^k	Factibilidade na iteração k para os métodos de otimização
$F_{ij}(\cdot)$	Forças gravitacionais do agente j no agente i no método GSA
G	Matriz de atenuação da fibra que conecta todos os pares transceptores
G_{ii}	Atenuação do sinal do caminho do código óptico
G_{ij}	Atenuação entre o j -ésimo nó transmissor e o i -ésimo nó receptor i
G	Constante gravitacional no método GSA
η_{EE}	Eficiência energética
η_{SE}	Eficiência espectral
H	Matriz de interferência normalizada
H_{ij}	ij -ésimo elemento da matriz de interferência normalizada
ϑ_i	Gap entre as taxas de informação teórica e real do i -ésimo usuário
$I_{\max}$	Número máximo de iterações
ι	Eficiência de energia do transponder $J_1(\cdot)$
$J_2(\cdot)$	Função custo para o controle de potência
$J_1^+(\cdot)$	Função J_1 redefinida
K	Número de dimensões do problema (usuários)
K_u	Número de usuários da classe de serviço u
$L(\cdot)$	Função Lagrangiana
λ_a	Número inteiro não negativo que representa a restrição na autocorrelação de códigos 2D
λ_c	Número inteiro não negativo que representa a restrição na correlação de códigos 2D
$\lambda_{\max}$	Maior autovalor da matriz H
Λ^*	Matriz diagonal com Γ^* alvo como seus elementos
M	Ordem de modulação
μ_i	Multiplicador de Lagrange para a i -ésima equação de restrição de desigualdade no ALM
μ_i^k	Multiplicador de Lagrange para a i -ésima equação de restrição de desigualdade na k -ésima iteração nos métodos SQP e ALM
M_{aj}	Massa gravitacional ativa relacionada agente j no método GSA
M_{pi}	Massa gravitacional passiva relacionada ao agente i no método GSA
N_T	Número de comprimento de ondas disponíveis
N_λ	Comprimento dos códigos 2D
ν	Incremento aplicado em ρ no ALM
p	Potência óptica de transmissão

p_i	Potência de transmissão do i -ésimo usuário
$p_{\min}$	Potência óptica mínima de transmissão
$p_{\max}$	Potência óptica máxima de transmissão
$\mathbf{p}^+$	Vetor $\mathbf{p}$ acrescido de variáveis de folga
p_{best_i}	Melhor posição do agente i na iteração k nas meta-heurísticas
P_T	Potência de transmissão total consumida
P_c	Potência do circuito
P	Operador projeção ortogonal
$rand$	Número aleatório com distribuição uniforme no intervalo $[0, 1]$ nas meta-heurísticas
r_c	Taxa de chip
r_i	Taxa de símbolos do i -ésimo usuário
$r_{i,\min}^{serv}$	Taxa de símbolos mínima de serviço para o i -ésimo usuário
R_{ij}	Distância euclidiana entre dois agentes i e j no método GSA
ρ	Parâmetro de penalidade do ALM
ρ_k	Parâmetro de penalidade na k -ésima iteração no ALM
T_b	Período de bit
T_c	Período de chip
T	Número total de realizações do processo
τ	Nível de decréscimo esperado em v^k no ALM
u	Classe de serviço
v_i^k	Velocidade da partícula i na iteração k nas meta-heurísticas
w	Peso do código
w_i	Largura de banda não espalhada do sinal do i -ésimo usuário
$\bar{w}$	Função de ponderação nas meta-heurísticas
W	Largura de banda do sistema óptico
x_i^k	Posição corrente da partícula i na iteração k das metaheurísticas
∇	Operador gradiente
∇_v	Operador gradiente em relação ao vetor v
∇^2	Operador matriz Hessiana

LISTA DE SIGLAS

ACO – *Ant Colony Optimization*
ALM – *Augmented Lagrangian method*
BCO – *Bee Colony Optimization*
BER – *Bit Error Rate*
CDMA – *Code Division Multiple Access*
CINR – *Carrier to Interference plus Noise Ratio*
EDFA – *Erbium Doped Fibre Amplifier*
EE – *Energy Efficiency*
EWHPC – *Extended Wavelength-Hopping Prime Codes*
FLOPS – *Floating-Point Operations Per Second*
GA – *Genetic Algorithm*
GSA – *Gravitational Search Algorithm*
MLEWHPC – *Multiple-length Extended Wavelength-hopping Prime Codes*
MIT – *Tree-phase Induction Motor*
NMSE – *Normalized Mean Square Error*
OCDMA – *Optical Code Division Multiple Access*
OLT – *Optical Line Termina*
ONU – *Optical Network Unite*
PID – *Proporcional, Integrativo, Derivativo*
PON – *Passive Optical Networks*
PSO – *Particle swarm optimization*
QoS – *Quality of Service*
RE – *Resource Efficiency*
SINR – *Signal to Interference plus Noise Ratio*

SUMÁRIO

1 INTRODUÇÃO	18
1.1 JUSTIFICATIVA	20
1.2 OBJETIVOS	21
1.3 CONTRIBUIÇÕES	22
1.4 ORGANIZAÇÃO DO TRABALHO	22
1.5 PUBLICAÇÕES CIENTÍFICAS ASSOCIADAS AO TRABALHO	23
1.5.1 Artigos Publicados em Periódicos	24
1.5.2 Artigo Sob Revisão em Periódico	24
1.5.3 Artigo Submetido em Periódico	24
1.6 CONCLUSÕES DO CAPÍTULO	24
2 FUNDAMENTAÇÃO TEÓRICA	25
2.1 VISÃO GERAL DA LITERATURA	25
2.2 OTIMIZAÇÃO	29
2.3 METAHEURÍSTICAS PARA OTIMIZAÇÃO	31
2.3.0.1 Intensificação e Diversificação	32
2.3.1 Técnicas para Aprimoramento da Metaheurística PSO	33
2.3.1.1 Distribuição Beta	33
2.3.1.2 Mapas Caóticos	35
2.3.1.3 Sequência de Halton	35
2.4 CALIBRAÇÃO DOS PARÂMETROS	39
2.5 ANÁLISE DAS METAHEURÍSTICAS	39
2.5.1 Critério de Parada	39
2.5.2 Robustez e Complexidade Computacional	40
2.6 FUNÇÕES BECHMARK	41
2.6.1 Função Ackley	41
2.6.2 Função Griewank	42
2.6.3 Função Rosenbrock	44
2.6.4 Função Rastrigin	44
2.7 CONCLUSÕES DO CAPÍTULO	46
3 METAHEURÍSTICAS UTILIZADAS	48
3.1 METAHEURÍSTICA DE OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS	48
3.2 METAHEURÍSTICA β -PSO	50
3.3 METAHEURÍSTICA CHAOS-PSO	51
3.4 HALTON-PSO	52
3.5 HALTON-CHAOS- β -PSO	53
3.6 RESULTADOS NUMÉRICOS DAS METAHEURÍSTICAS VALIDADAS COM FUNÇÕES BENCHMARK	55
3.6.1 Calibração dos Parâmetros das Metaheurísticas para as Funções Benchmark	56
3.6.1.1 Calibração dos Parâmetros da Metaheurística PSO Tradicional	56
3.6.1.2 Calibração dos Parâmetros da Metaheurística β -PSO	58
3.6.1.3 Calibração dos Parâmetros da Metaheurística Chaos-PSO	68

3.6.1.4 Calibração dos Parâmetros da Metaheurística Halton-PSO	72
3.7 COMPARAÇÃO ENTRE AS METAHEURÍSTICAS	77
3.7.1 Simulações e Resultados	78
3.7.2 Análise da Complexidade Computacional	86
3.8 CONCLUSÕES DO CAPÍTULO	91
4 METAHEURÍSTICA HALTON-CHAOS-β-PSO PARA ESTIMAÇÃO DE PARÂMETROS	
DE UM MOTOR DE INDUÇÃO TRIFÁSICO	97
4.1 IDENTIFICAÇÃO DE PARÂMETROS DO MIT	98
4.1.1 Modelo do MIT	98
4.1.2 Problema de Estimação de Parâmetros do MIT	100
4.2 METAHEURÍSTICAS PARA ESTIMAÇÃO DE PARÂMETROS DO MIT	102
4.2.1 Algoritmos ALM e ALPSO	102
4.2.2 Algoritmo PSOGSA	103
4.3 CALIBRAÇÃO DOS PARÂMETROS DE ENTRADA	104
4.3.1 Calibração dos Parâmetros de Entrada – PSO Tradicional	105
4.3.2 Calibração dos Parâmetros de Entrada – β -PSO	106
4.3.3 Calibração de Parâmetros de Entrada – Chaos-PSO	106
4.3.4 Calibração dos Parâmetros de Entrada – Halton-PSO	107
4.3.5 Calibração dos Parâmetros de Entrada do Algoritmo Híbrido PSOGSA	108
4.4 COMPARAÇÃO ENTRE OS MÉTODOS DE ESTIMAÇÃO DE PARÂMETROS ..	108
4.4.1 Critério de Convergência	108
4.4.2 Simulações e Resultados	109
4.4.3 Complexidade Computacional	112
4.5 CONCLUSÕES DO CAPÍTULO	114
5 METAHEURÍSTICA HALTON-CHAOS-β-PSO NA ALOCAÇÃO DE POTÊNCIA	
EM REDES PON-OCDMA	115
5.1 PROBLEMA DE ALOCAÇÃO DE POTÊNCIA EM SISTEMAS PON-OCDMA	115
5.1.1 Alocação de Taxa de Potência	116
5.2 RESULTADOS NUMÉRICOS	118
5.2.1 Calibração dos Parâmetros de Entrada da Metaheurística PSO	119
5.2.2 Calibração dos Parâmetros de Entrada – Metaheurística β -PSO	121
5.2.3 Calibração dos Parâmetros de Entrada – Chaos-PSO	122
5.2.4 Calibração dos Parâmetros de Entrada – Halton-PSO	122
5.3 COMPARAÇÃO ENTRE OS MÉTODOS DE ALOCAÇÃO DE TAXA DE POTÊNCIA	
123	
5.3.1 Simulações e Resultados	123
5.3.2 Complexidade Computacional	126
5.4 CONCLUSÕES DO CAPÍTULO	131
6 CONCLUSÕES	132
6.1 TRABALHOS FUTUROS	133
REFERÊNCIAS	134

1 INTRODUÇÃO

A otimização é relacionada ao estudo de problemas em que se busca maximizar ou minimizar uma função através da escolha sistemática dos valores de variáveis dentro de um conjunto factível. As formas clássicas ou exatas de otimização, como os métodos iterativos e numéricos, não são eficazes e flexíveis para lidar com problemas dinâmicos, não lineares ou que apresentem restrições (FOULDS, 2012). Para contornar essas limitações, atualmente são empregadas técnicas inteligentes, as quais apresentam bom compromisso entre complexidade e custo (JONG, 2016); (ENGELBRECHT, 2007). Dentre as técnicas inteligentes de otimização, tem-se as metaheurísticas bioinspiradas, que tentam simular o comportamento evolutivo e adaptativo dos seres vivos, sendo frequentemente inspiradas na natureza e estão entre as técnicas atuais mais utilizadas para a resolução de problemas de otimização complexos ou não lineares (YANG, 2014); (JONG, 2016).

As metaheurísticas apresentam alguns parâmetros de projeto e o cálculo de uma função custo, que deve apresentar um baixo custo computacional (YANG, 2014). A maioria das metaheurísticas é baseada em uma população de indivíduos que apresentam efetividade para a solução de problemas de otimização complexos. Apesar de não se poder garantir a exatidão dos resultados obtidos em todos os casos, quando comparado aos métodos exatos, os resultados alcançados são quase ótimos, possuindo complexidade computacional inferior aos métodos tradicionais (YANG, 2010). Como exemplos de metaheurísticas bioinspiradas muito utilizadas na literatura, tem-se: Otimização por Enxame de Partículas (PSO) (EBERHART; KENNEDY, 1995), Otimização por Colônia de Formigas (ACO) (DORIGO; CARO, 1999), Algoritmo Genético (GA) (HOLLAND, 1984), Algoritmo por Colônia de Abelhas (BCO) (KARABOGA; OZTURK, 2011), entre outros.

A metaheurística de Otimização por Enxame de Partículas é amplamente utilizada na área de Engenharia Elétrica, principalmente para a estimação de componentes harmônicos em sistemas elétricos de potência (PRINCE et al., 2018); (SRIVASTAVA et al., 2018); (SREEKUMAR; TRAD, 2018), rastreamento do ponto máximo de potência em arranjos fotovoltaicos (BALAMURUGAN et al., 2016); (ÑAUPARI et al., 2018); (GONG et al., 2017), sintonia de

controladores PID (ZAHRA et al., 2017); (AWASTHI et al., 2016), alocação de potência em redes de fibra óptica (SANTANA et al., 2018); (DURAND; ABRÃO, 2017), controle e estimação de parâmetros de máquinas elétricas (CUEVAS et al., 2019); (SONG et al., 2018) (ZHENG; LIAO, 2016) etc.

O PSO é um algoritmo estocástico inspirado no comportamento dos pássaros e utilizado para otimizar um problema através da busca iterativa por um valor máximo ou mínimo, baseado em uma função custo (EBERHART; KENNEDY, 1995). A técnica PSO é baseada no movimento aleatório de um enxame (população) de partículas (indivíduos), que possuem velocidade e posição, e se movimentam pelo espaço de busca a procura de uma solução ótima para um determinado problema, sendo que cada partícula representa uma potencial solução (EBERHART; KENNEDY, 1995); (BOUSSAÏD et al., 2013). O movimento de cada partícula depende de sua melhor posição individual encontrada até o momento, bem como da melhor posição global encontrada pelo enxame. A medida que posições mais promissoras são encontradas, essa informação serve para guiar o movimento das partículas através do espaço de busca.

Assim como em todas as metaheurísticas bioinspiradas, o principal desafio do PSO é conseguir o equilíbrio adequado entre as fases de intensificação e diversificação no processo de busca para um problema específico de otimização (YANG, 2010). Além disso, a grande desvantagem do PSO está associada ao fato de que o algoritmo pode facilmente ficar preso em ótimos locais quando o espaço de busca possui alta dimensão, ocasionando a convergência prematura e, conseqüentemente, afetando a qualidade das soluções encontradas (BLUM et al., 2012).

Considerando essas limitações do algoritmo PSO, esta Tese propõe uma nova metaheurística baseada na otimização por enxame de partículas, denominada Halton-Chaos- β -PSO. A metaheurística proposta utiliza a Distribuição Beta (MACA; PECH, 2015); (ALI, 2007) ao invés da distribuição uniforme, utilizada pelo PSO tradicional, e um fator de amortecimento baseado em mapas caóticos (YANG et al., 2014); (ALATAS et al., 2009) para a atualização do melhor valor global encontrado no processo de otimização, aprimorando a capacidade de busca global do algoritmo ao escapar das soluções locais. Além disso, para a inicialização do enxame de partículas é usada a sequência de Halton (DIGEHSARA et al., 2020); (STEINERBERGER, 2020) ao invés da inicialização aleatória utilizada pelo algoritmo tradicional. A agregação dessas características tem como objetivo alcançar um satisfatório compromisso entre desempenho e complexidade, além de proporcionar maior velocidade de convergência e qualidade nas soluções.

Para a validação da metaheurística Halton-Chaos- β -PSO são utilizadas funções Benchmark, comparando seu desempenho ao obtido pelo PSO tradicional, em termos de velocidade de convergência, erro quadrático médio normalizado e número de FLOPs. Para todas as metaheurísticas comparadas é investigada a sintonia ótima de parâmetros com o objetivo da melhoria da qualidade das soluções encontradas. Além disso, a metaheurística proposta é aplicada em dois problemas de otimização da área de Engenharia Elétrica: alocação de potência em redes de fibra ótica e estimação de parâmetros de um motor de indução trifásico.

1.1 JUSTIFICATIVA

Muitos problemas de otimização não podem ser tratados por meio de métodos analíticos ou exatos, principalmente por conta da dificuldade da formulação da modelagem matemática exigida na solução ou quando funções não-diferenciáveis ou descontínuas estão envolvidas (COLLATZ; WETTERLING, 2012). Grande parte dos problemas da área da engenharia é combinatória, sendo os mesmos classificados como NP1-Difícil ou NP-Completo (ODILI, 2017). Segundo Parpinelli et al. (2019), um problema é considerado intratável ou NP-Completo se não existirem algoritmos capazes de resolvê-lo em tempo polinomial. Para tais tipos de problemas ainda não existem algoritmos capazes de produzir soluções ótimas em tempo polinomial (COLLATZ; WETTERLING, 2012).

Nesse contexto, são necessários métodos alternativos que proporcionem uma solução próxima daquela que seria a ótima e que apresentem um bom compromisso entre desempenho e complexidade. Nesse contexto, faz-se uso de métodos de aproximação, randomização e, principalmente, técnicas metaheurísticas (BOUSSAÏD et al., 2013). As metaheurísticas, quando bem aplicadas, podem resultar em um conjunto de soluções de boa qualidade e com baixo custo computacional (YANG, 2010); (JONG, 2016).

Nos últimos anos, metaheurísticas bioinspiradas baseadas em populações vêm sendo utilizadas para a resolução de problemas complexos de otimização e têm apresentado desempenho relativamente bom em vários domínios de aplicação, inclusive em complexos, demonstrando habilidade de adaptação, flexibilidade e boa generalização (YANG, 2010); (JONG, 2016); (BLUM et al., 2012); (PARPINELLI et al., 2019).

No entanto, as metaheurísticas possuem algumas desvantagens e seu desempenho está atrelado à calibração dos parâmetros de entrada utilizados para a realização do processo de otimização de um problema específico (COLLATZ; WETTERLING, 2012). Além disso, para o bom funcionamento das metaheurísticas, deve-se encontrar o equilíbrio adequado entre as

fases de exploração e intensificação, caso contrário, o algoritmo pode apresentar convergência prematura, parando o processo de busca ao encontrar um ponto ótimo local, e assim, comprometendo o desempenho e as soluções encontradas pelo algoritmo metaheurístico.

Considerando esse contexto, justifica-se o desenvolvimento de uma nova abordagem metaheurística para a resolução de problemas complexos que apresente um bom compromisso entre desempenho e custo computacional, além de garantir boas soluções e fornecer rápida convergência. A metaheurística proposta Halton-Chaos- β -PSO agrega algumas características relevantes ao PSO tradicional com o objetivo de oferecer maior rapidez na convergência, qualidade nos resultados obtidos e compromisso entre desempenho e complexidade. O algoritmo Halton-Chaos- β -PSO utiliza a distribuição Beta e um fator de amortecimento baseado em mapas caóticos para evitar a convergência prematura, no caso do algoritmo encontrar ótimos locais no espaço de busca, aprimorando o processo de exploração e intensificação, além de utilizar um tipo de sequência de Monte Carlo para a inicialização do enxame de partículas, o que pode proporcionar melhor capacidade de exploração do espaço de busca e menor tempo para encontrar o ótimo global de um problema de otimização.

1.2 OBJETIVOS

O objetivo geral desta Tese é o desenvolvimento de uma nova metaheurística bioinspirada por enxame de partículas para a resolução de problemas de otimização. Ao objetivo geral deste trabalho estão relacionados os seguintes objetivos específicos:

- Investigar as principais metaheurísticas bioinspiradas presentes na literatura;
- Verificar o funcionamento das metaheurísticas mais difundidas;
- Investigar as principais técnicas de calibração de parâmetros de entrada para essas metaheurísticas;
- Verificar o compromisso entre desempenho e complexidade das metaheurísticas mais utilizadas;
- Investigar as principais desvantagens da metaheurística de otimização por enxame de partículas;
- Investigar as principais técnicas de melhorias utilizadas para o aprimoramento do desempenho da metaheurística de otimização por enxame de partículas;
- Desenvolver uma nova metaheurística baseada na otimização por enxame de partículas;

- Calibrar os parâmetros de entrada da nova metaheurística;
- Verificar o desempenho e complexidade da nova metaheurística através de funções Benchmark;
- Comparar a metaheurística proposta, em termos do compromisso entre complexidade e desempenho, ao algoritmo PSO tradicional;
- Aplicar e validar a metaheurística desenvolvida através da solução de problemas práticos da área de Engenharia Elétrica.

1.3 CONTRIBUIÇÕES

A proposta deste trabalho busca contribuir na construção de resultados de pesquisas relevantes para o desenvolvimento de novas metaheurísticas de otimização bioinspiradas. Justificando a necessidade de otimização desses recursos, os seguintes aspectos foram explorados nesta Tese:

1. Distribuição Beta: utilizada ao invés da distribuição uniforme, para a geração de variáveis aleatórias com o objetivo de aumentar a diversidade e a exploração de regiões não exploradas no espaço de busca durante o processo de otimização;
2. Fator de amortecimento baseado em mapas caóticos: responsável pela geração de números aleatórios usados para a atualização do melhor valor global encontrado pela metaheurística. Essa característica pode limitar a dominância do valor da melhor partícula e evitar a convergência prematura, aumentando a aleatoriedade (diversificação) sem perda na capacidade de exploração;
3. Sequência de Halton: usada para aprimorar a qualidade da geração da população inicial na primeira etapa, objetivando a cobertura apropriada do espaço factível de busca.

1.4 ORGANIZAÇÃO DO TRABALHO

Esta Tese de Doutorado contém, além do capítulo introdutório, seis partes, descritas a seguir:

- **Capítulo 2** - Neste Capítulo, é realizada uma visão geral da literatura, destacando os principais trabalhos referentes à metaheurística PSO, assim como técnicas utilizadas para

a melhoria do desempenho deste algoritmo. Além disso, são apresentados alguns trabalhos que aplicam o PSO para a resolução de problemas de otimização na área de Engenharia Elétrica. Adicionalmente, este Capítulo apresenta os conceitos teóricos fundamentais sobre otimização e metaheurísticas bioinspiradas. Adicionalmente, são discutidas as técnicas de calibração de parâmetros utilizadas neste trabalho, e aspectos fundamentais para a implementação de metaheurísticas, como: critério de convergência, complexidade computacional e robustez algorítmica. Por fim, são apresentadas as funções Benchmark que servem como base para a validação do algoritmo Halton-Chaos- β -PSO.

- **Capítulo 4** - Neste Capítulo, são apresentadas as metaheurísticas utilizadas por este trabalho, especialmente a metaheurística proposta Halton-Chaos- β -PSO, assim como é realizada a validação desses algoritmos utilizando funções Benchmark. Os resultados do compromisso entre desempenho e complexidade da abordagem proposta são discutidos e comparados ao PSO tradicional.
- **Capítulo 5** - Neste Capítulo, é apresentada a aplicação da metaheurística H-C- β -PSO para resolver o problema de estimação de parâmetros de um motor de indução trifásico sujeito à mudança de parâmetros dessa máquina durante sua operação. Além disso, a abordagem proposta é comparada com o PSO tradicional e algumas de suas variantes, assim como é comparado com uma técnica analítica e um método híbrido, destacando os resultados obtidos em relação ao compromisso desempenho-complexidade.
- **Capítulo 6** - Neste Capítulo, é realizada a aplicação e comparação da capacidade, eficácia e eficiência do algoritmo Halton-Chaos- β -PSO com o PSO tradicional, versões melhoradas desse algoritmo e o método ALM-PSO, para a resolução do problema de alocação de potência em redes óticas PON-OCDMA.
- **Capítulo 7** - Finalmente, neste Capítulo são apresentadas as principais conclusões e sugestões para continuidade dos estudos.

1.5 PUBLICAÇÕES CIENTÍFICAS ASSOCIADAS AO TRABALHO

Os artigos científicos/tecnológicos gerados como resultado do trabalho de investigação desta Tese de doutorado são sintetizados a seguir.

1.5.1 ARTIGOS PUBLICADOS EM PERIÓDICOS

- [A] SANTANA, Gisele A.; DURAND, Fábio R.; ABRÃO, Taufik. Power Allocation in PON-OCDMA with Improved Chaos Particle Swarm Optimization. **Journal of Microwaves, Optoelectronics and Electromagnetic Applications**, v. 17, n. 2, p. 268-283, 2018; DOI: <https://doi.org/10.1590/2179-10742018v17i21258>, ISSN 2179-1074, IF: [n.a.] Available online: 30 September 2018.
URL: <http://www.jmoe.org/index.php/jmoe/article/view/673>

1.5.2 ARTIGO SOB REVISÃO EM PERIÓDICO

- [B] SANTANA, G. A.; PENDEZA MARTINEZ, C. A.; MARTINEZ, A. L. M.; SANTOS, L.; ABRÃO, T., **Halton-Chaos and ALM-PSO Power Allocation Methods for Passive Optical CDMA Networks**. *AEUE - International Journal of Electronics and Communications*. Elsevier. ISSN: 1434-8411
IF: 3.183 Versão revisada **R1** em 18 de junho de 2021.
URL: <https://www.journals.elsevier.com/aeu-international-journal-of-electronics-and-communications>

1.5.3 ARTIGO SUBMETIDO EM PERIÓDICO

- [C] SANTANA, G. A.; PENDEZA MARTINEZ, C. A.; MARTINEZ, A. L. M.; GOEDTEL, A.; ABRÃO, T. **Parameter Identification in Three-phase Induction Motor Aided by Halton-Chaos- β -PSO Algorithm**. *Soft Computing*. Springer
IF: 3.050 Versão original **R0** submetida em 8 de junho de 2021.
URL: <https://www.springer.com/journal/500>

1.6 CONCLUSÕES DO CAPÍTULO

Neste capítulo, foram apresentados os conceitos principais e objetivos desenvolvidos nesta Tese de doutorado, bem como elencadas as justificativas para o desenvolvimento da investigação científica-tecnológica e a organização dos capítulos. Por fim, foram apresentadas as principais produções bibliográficas resultantes deste trabalho de Tese.

2 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo, é apresentada uma visão geral da literatura, procurando mostrar o cenário atual do campo de metaheurísticas, sendo apresentado um estudo das principais publicações entre os anos de 2017 e 2021. Além disso, é realizada a revisão dos principais trabalhos que aplicam a metaheurística PSO à problemas da área de Engenharia Elétrica. Este Capítulo também apresenta os conceitos teóricos fundamentais sobre metaheurísticas e otimização. Adicionalmente, é abordada a técnica de calibração dos parâmetros de entrada e as técnicas empregadas para a melhoria da metaheurística PSO utilizadas neste trabalho. Além disso, são apresentadas as funções de teste (Benchmark) usadas para a validação da metaheurística proposta.

2.1 VISÃO GERAL DA LITERATURA

No trabalho de Bansal (2019) é observado que algoritmos metaheurísticos apresentam comportamento altamente não linear, complexo e estocástico. Assim, a maioria dos trabalhos realiza a análise de desempenho das metaheurísticas de maneira quantitativa, utilizando métodos de validação, como: desvio padrão, média, erro quadrático médio normalizado, etc, como realizado em Parsopoulos e Vrahatis (2019). Além disso, para essa análise são utilizadas funções denominadas Benchmark, sendo que a maioria das novas metaheurísticas compara seus resultados aos obtidos pelas técnicas bioinspiradas tradicionais, como o PSO (WANG et al., 2018); (SENGUPTA et al., 2019); (KIRAN, 2017). Nesse contexto, Sengupta et al. (2019) destaca que a análise de convergência e eficiência de novas metaheurísticas ainda é um fator que está em aberto à novas técnicas e pesquisas.

Em relação aos parâmetros de entrada dos algoritmos metaheurísticos, Harrison et al. (2019) observa que a definição de bons parâmetros influencia fortemente na convergência dos algoritmos e que a calibração desses parâmetros é um processo trabalhoso, já que depende especificamente do problema de otimização analisado. Nos trabalhos de Harrison et al. (2018)

e Sedighizadeh et al. (2021), o ajuste dos parâmetros de entrada se mostrou como um ponto interessante para direções futuras de pesquisa, já que em termos de análise de convergência, complexidade e tempo de execução, esses parâmetros possuem grande influência. Porém, nesses trabalhos não é demonstrada a metodologia utilizada para a calibração dos parâmetros. Já em Pedersen (2020) é realizada uma análise para a definição de bons parâmetros de entrada para o algoritmo PSO, considerando diferentes dimensões para algumas funções Benchmark, porém também não é explicado o processo de ajuste dos parâmetros.

Em relação às técnicas de randomização utilizadas pelas metaheurísticas, as mais comuns são: distribuições uniformes simples ou métodos complexos, como Monte Carlo, mapas caóticos ou distribuição Beta. Nos trabalhos de Yang et al. (2014); Araujo e Coelho (2008) e Pluhacek et al. (2015), alguns mapas caóticos são agregados ao algoritmo PSO para a geração de variáveis aleatórias para a atualização da velocidade das partículas e também para encontrar novas soluções na vizinhança das melhores posições já encontradas pelo algoritmo. Já a distribuição Beta foi previamente empregada para melhorar a metaheurística PSO em relação à atualização do peso inercial (MACA; PECH, 2015) e inicialização da velocidade das partículas (ANDRÉ et al., 2015).

Por sua vez, no trabalho de Digehsara et al. (2020) a metaheurística Halton-PSO (H-PSO) utiliza sequências de Halton, um tipo de sequência de Monte Carlo, como mecanismo para a geração da população inicial na primeira etapa do processo de otimização, objetivando apropriada cobertura do espaço viável de busca. Através da utilização da sequência de Halton Embaralhada, responsável pela geração de números quase aleatórios, a população inicial do algoritmo é gerada de maneira mais eficiente sem afetar diretamente a estrutura do algoritmo PSO tradicional.

De acordo com Rezaei e Safavi (2020), os algoritmos metaheurísticos apresentam dois componentes fundamentais: intensificação e exploração do processo de busca, sendo que o grande desafio é a implementação de um algoritmo que encontre o equilíbrio entre esses componentes. De acordo com (ŞENEL et al., 2019), o entendimento sobre como esses dois fatores afetam o desempenho de qualquer metaheurística ainda possui limitações. Geralmente, as abordagens mais utilizadas para equilibrar intensificação e exploração estão relacionadas ao controle do tamanho da população (RAHMAN et al., 2020), calibração dos parâmetros de entrada (REZAEI; SAFAVI, 2020); (HARRISON et al., 2019) e manutenção da diversidade através de técnicas determinísticas ou adaptativas (HARRISON et al., 2018); (KIRAN, 2017).

Na área de Engenharia Elétrica, muitas aplicações utilizam a metaheurística PSO para a resolução de problemas, pois essa metaheurística apresenta bons resultados quando

comparado a outras heurísticas tradicionais. Nessa área, o PSO é principalmente utilizado para a estimação de componentes harmônicos em sistemas elétricos de potência, rastreamento do ponto máximo de potência em arranjos fotovoltaicos, sintonia de controladores PID, alocação de potência em redes de fibra ótica, projeto e análise de custo de geradores eólicos, controle e estimação de parâmetros de máquinas elétricas etc.

No trabalho de Medjghou et al. (2018) é apresentado um filtro de Kalman baseado na estratégia de controle proporcional integral (PI) para controlar o filtro de potência ativa em derivação. Para obter precisão no desempenho do filtro, o algoritmo PSO é adotado para o ajuste dos parâmetros, apresentado bons resultados finais. Em Srivastava et al. (2018), um controlador PI de ordem fracionária é projetado para melhorar o desempenho do filtro de potência ativa em derivação mantendo uma tensão constante. Os parâmetros do controlador são ajustados utilizando a otimização por enxame de partículas. Os resultados obtidos são comparados com um controlador PI geralmente usado para a mesma aplicação, e mostram que o estudo proposto fornece melhores resultados na manutenção do nível de tensão e na redução dos harmônicos.

No trabalho de Lee et al. (2017), é proposto um algoritmo PSO modificado aplicado ao projeto ótimo de um motor síncrono de ímã permanente, visando minimizar a distorção harmônica total da força contra-eletromotriz. Para melhorar o desempenho do algoritmo PSO tradicional, um novo método é aplicado para controlar de maneira inteligente o número de partículas. O novo método compara o valor de custo do melhor global na iteração atual com o valor na iteração anterior. Se houver diferença entre os dois valores de custo, o algoritmo proposto opera no estágio de exploração, mantendo o número de partículas. No entanto, quando a diferença nos valores de custo é menor que os valores de tolerância atribuídos pelo usuário, o algoritmo proposto opera no estágio de intensificação, reduzindo o número de partículas. Além disso, o algoritmo elimina a partícula que está mais próxima da melhor valor global para preservar a aleatoriedade.

Nos trabalhos de Bhowmick et al. (2018) e Bhowmick et al. (2016b) foi proposta uma técnica simples e robusta para a estimação de parâmetros do circuito equivalente de um motor de indução trifásico a partir de dados obtidos experimentalmente em condições de operação. Os valores de tensão, corrente, potência e velocidade foram medidos sob diferentes condições de carga e os dados foram analisados utilizando a otimização por enxame de partículas juntamente com a técnica de estimativa de resistência. Os parâmetros estimados foram comparados com os valores correspondentes obtidos através do procedimento de teste clássico. O estudo comparativo mostrou que a técnica proposta pode ser utilizada de forma confiável para estimar

os parâmetros de máquinas elétricas.

Em Sahoo et al. (2019), a metaheurística PSO foi utilizada como técnica para rastrear o ponto máximo de potência de um arranjo fotovoltaico. Nesse trabalho, o objetivo principal do PSO foi descobrir o ciclo de trabalho do conversor Boost para manter a tensão de saída constante, independentemente da energia produzida pelos painéis solares. Em Gong et al. (2017) foi proposto um novo método baseado no algoritmo PSO para determinar os parâmetros desconhecidos do modelo do arranjo fotovoltaico. A técnica PSO foi aplicada para a identificação dos parâmetros do arranjo e adotou algumas melhorias no processo iterativo, com a finalidade de garantir a convergência efetiva, utilizando peso de inércia adaptativo e fatores de aprendizagem variáveis.

No trabalho de Ren e Liu (2017), um algoritmo modificado de otimização por enxame de partículas foi desenvolvido para resolver problemas de otimização na Engenharia Elétrica. O mecanismo de aprendizagem das partículas foi usado para guiar a evolução da população em regiões viáveis e o algoritmo utilizou o peso inercial adaptativo e fatores de aprendizagem para equilibrar a exploração do espaço de busca. A estratégia de redefinição de partículas foi utilizada para que o algoritmo encontrasse o ótimo global de forma eficaz. Nesse trabalho, a população inicial foi gerada a partir da teoria do caos.

Em Santana et al. (2018) é proposto um algoritmo de alocação de potência para redes PON-OCDMA combinando a otimização por enxame de partículas tradicional e a teoria de caos, denominado IC-PSO, tendo em vista melhorar a velocidade de convergência e a qualidade das soluções do PSO tradicional. O IC-PSO proposto utilizou a distribuição Beta ao invés da distribuição uniforme para a atualização da velocidade das partículas; além disso, um fator de amortecimento baseado no mapa logístico caótico relacionado à atualização do melhor valor global foi introduzido com sucesso. Essas características afetaram de maneira positiva o compromisso desempenho-complexidade, a velocidade de convergência e a qualidade das soluções obtidas.

Como observado, a metaheurística PSO é amplamente empregada para a otimização de problemas complexos, inclusive na área de Engenharia Elétrica, pois possui fácil implementação, bom desempenho e, principalmente, vários aspectos que podem ser melhorados. Assim, este trabalho propõe o desenvolvimento de uma nova abordagem metaheurística, denominada Halton-Chaos- β -PSO, que é baseada no algoritmo tradicional de otimização por enxame de partículas, mas utiliza algumas técnicas de calibração de parâmetros de entrada, atualização de velocidade e geração da população inicial para melhorar o compromisso entre complexidade e desempenho do PSO tradicional. Para validar a abordagem proposta,

o algoritmo é aplicado para a resolução de dois problemas de otimização da área de Engenharia Elétrica: estimação de parâmetros de um motor de indução trifásico e alocação de potência em redes óticas PON-OCDMA. As metaheurísticas podem resolver uma ampla classe de problemas complexos de otimização, obtendo resultados superiores em termos do compromisso desempenho-complexidade em relação às técnicas determinísticas, quer sejam exatas ou analíticas-iterativas. Apesar de existir uma vasta literatura relacionada ao desenvolvimento e aplicação de metaheurísticas para a resolução de problemas de otimização, observou-se que ainda há espaço para pesquisa de novas técnicas ou aperfeiçoamentos de técnicas de diversificação (*randomização*) para a geração da população inicial de metaheurísticas populacionais, assim como a melhoria do compromisso entre desempenho e complexidade, melhoria do processo de intensificação e diversificação, aprimoramento do processo de calibração dos parâmetros de entrada das metaheurísticas, entre outros aspectos de projeto de algoritmos heurístico-evolucionários.

2.2 OTIMIZAÇÃO

A otimização é uma área da matemática destinada à obtenção de condições que proporcionem o valor extremo de uma ou várias funções sob determinadas condições (COLLATZ; WETTERLING, 2012). Segundo Odili (2017), a otimização é utilizada para maximizar ou minimizar o resultado de um problema com recursos limitados, selecionando a melhor solução dentre um conjunto de soluções disponíveis. Dessa maneira, a otimização pode ser definida como um processo de se encontrar o valor máximo ou mínimo de uma função de acordo com determinadas restrições (YANG, 2010).

Formalmente, um problema de otimização Optz pode ser definido por:

$$\text{Optz} = (\mathcal{S}, \mathcal{V}, \nu, f) \quad (1)$$

no qual $\mathcal{S}$ é o espaço de busca de um conjunto finito de variáveis de decisão $\nu = \{1, \dots, n\}$, $\mathcal{V}$ é o conjunto de restrições entre as variáveis de decisão, e f é a função custo a ser otimizada. Para a resolução do problema, uma solução ótima s^* deve ser encontrada com o valor mínimo da função custo $f(s^*) \leq f(s), \forall s \in \mathcal{S}$, ou no caso de maximização: $f(s^*) \geq f(s), \forall s \in \mathcal{S}$.

O espaço de busca $\mathcal{S}$ é um conjunto que contém todas as soluções do problema, podendo ser finito ou infinito enumerável, considerando apenas que esse espaço contenha apenas soluções viáveis, isto é, aquelas que obedecem às restrições do problema (se houver). Graficamente, conforme ilustrado na Figura 1, se um ponto x^* corresponde ao valor mínimo da função $f(x)$, o mesmo ponto também corresponde ao valor máximo do negativo da função,

$-f(x)$.

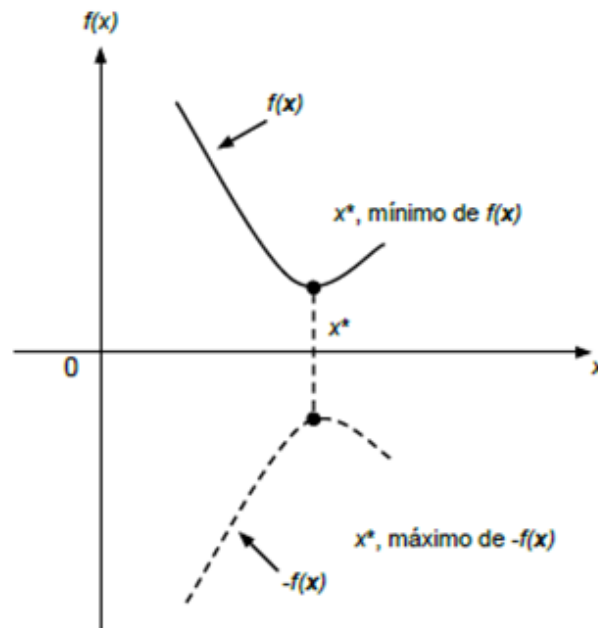


Figura 1: Pontos máximo e mínimo de uma função.

Fonte: Adaptado de (BOUSSAÏD et al., 2013)

Os problemas de otimização podem ser de vários tipos: sem restrições, com restrições ou combinatoriais. Muitos problemas de otimização possuem certas restrições ou regras que não podem ser violadas ao se encontrar uma solução otimizada. Portanto, um problema com restrições envolve a função custo $f(x)$ que deve ser minimizada ou maximizada com base em certas restrições. Já os problemas de otimização irrestritos são aqueles que não envolvem restrições ou limitações e dependem de variáveis reais. Tais problemas possuem uma função custo que minimiza ou maximiza diretamente o valor de $f(x)$ (COLLATZ; WETTERLING, 2012).

Existem várias técnicas de otimização que podem ser desenvolvidas através de métodos tradicionais, como a programação linear, programação não linear ou algoritmos baseados na inteligência de enxames. Geralmente, os métodos clássicos são classificados como métodos diretos ou gradiente, com base na técnica de busca usada para encontrar o melhor valor para um determinado problema de otimização. Os métodos diretos usam o valor de uma função custo, enquanto o método gradiente utiliza informações da função derivada para guiar a busca (BOUSSAÏD et al., 2013). Entretanto, os métodos tradicionais são mais adequados para funções custo simples e unimodais, não sendo eficazes para problemas complexos multimodais (FOULDS, 2012). Nesses casos, a fim de obter o resultado eficiente, são desenvolvidas técnicas não tradicionais, também conhecidas por heurísticas. Os métodos heurísticos podem

ser classificados como evolutivos, naturais e baseados em técnicas lógicas, conforme Figura 2.

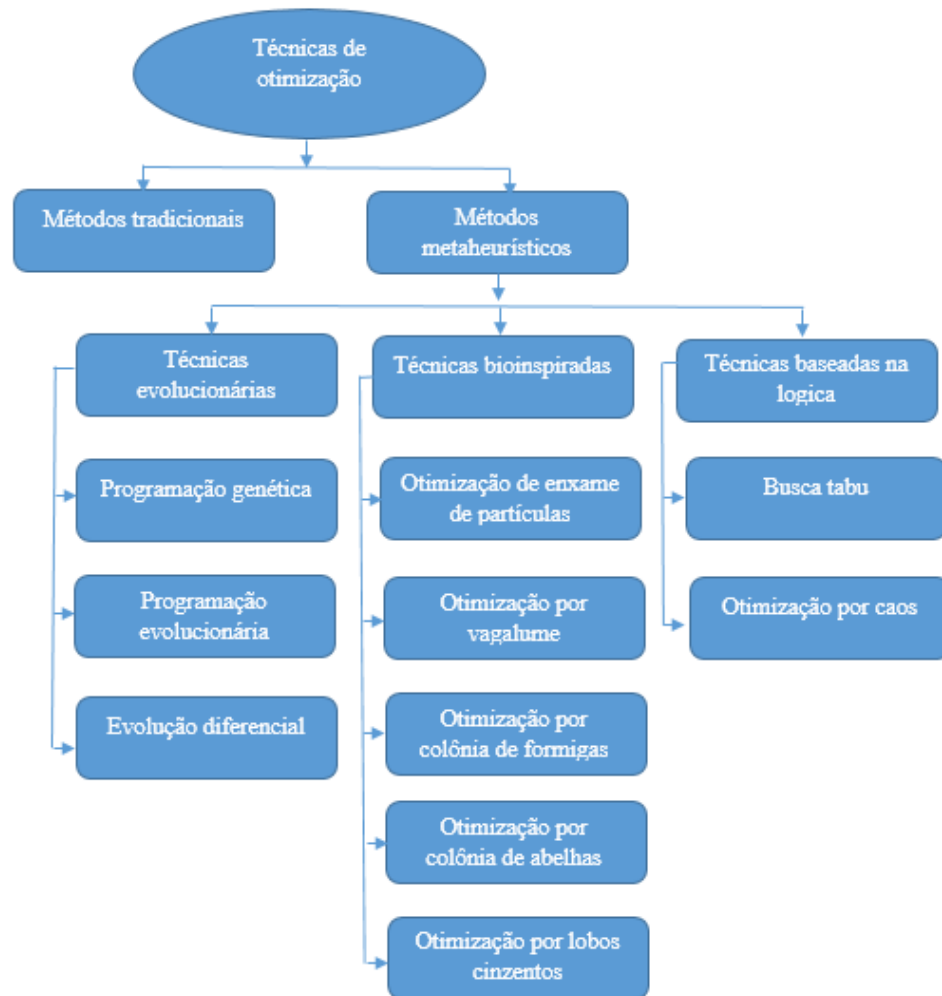


Figura 2: Classificação dos Métodos Heurísticos.

Fonte: Adaptado de (YANG, 2010)

Na próxima seção, são apresentados os conceitos fundamentais sobre metaheurísticas utilizadas para otimização.

2.3 METAHEURÍSTICAS PARA OTIMIZAÇÃO

Uma metaheurística é um método utilizado para resolver de forma genérica problemas de otimização. Geralmente são aplicadas a problemas para os quais não se conhece algoritmo eficiente, como problemas NP-completos ou difíceis, por oferecerem melhores soluções e geralmente com tempo de processamento menor que outros tipos de técnicas, como as determinísticas (YANG, 2010). As metaheurísticas são usadas para resolver problemas de otimização pelo processo de busca de soluções ótimas para um problema de interesse específico.

O processo de busca pode ser realizado por meio de vários agentes que formam essencialmente um conjunto de soluções em constante evolução, usando um conjunto de regras ou equações matemáticas durante as iterações. Essas iterações são mantidas até que a solução encontrada atenda a algum critério predefinido, como o número máximo de iterações ou a tolerância mínima. A solução final, também conhecida por solução quase ótima, é considerada uma solução ótima e considera-se que o algoritmo atingiu um estado convergente (BANSAL, 2019).

Os algoritmos metaheurísticos encontram a solução quase ótima rapidamente, ao contrário dos métodos exatos ou determinísticos que encontram essa solução com maior tempo computacional. As técnicas metaheurísticas obtiveram grande sucesso, pois fornecem soluções com um custo computacional aceitável (YANG, 2014).

As metaheurísticas bioinspiradas fazem parte do conjunto de algoritmos metaheurísticos e utilizam a inteligência evolutiva de processos biológicos com o objetivo de resolver problemas de difícil solução, cujo o espaço de busca possui grande dimensão (JONG, 2016). Essas metaheurísticas não garantem a solução ótima, mas permitem uma avaliação mais eficiente do espaço de busca, obtendo, na maioria das vezes, soluções com melhor qualidade (YANG, 2014). Geralmente, são aplicadas em situações nas quais as variáveis de projeto interagem entre si de maneira complexa, ou em problemas nos quais existe a possibilidade de se encontrar vários ótimos locais ou soluções sub ótimas (BLUM et al., 2012).

Os algoritmos bioinspirados são muito similares, apenas mudando alguns elementos, como a representação e avaliação dos dados de entrada e as operações sobre esses dados. Em sua maioria, esses algoritmos compartilham algumas características como a utilização de componentes estocásticos e os parâmetros de entrada (YANG, 2014). Cada tipo de algoritmo possui uma quantidade específica de parâmetros a ser ajustada e esse ajuste controla a execução do algoritmo, afeta diretamente a qualidade da solução final e influencia o direcionamento da busca por regiões promissoras no espaço de soluções do problema. No desenvolvimento de uma metaheurística, alguns critérios devem ser considerados, como: busca no espaço de soluções (diversificação), exploração das melhores soluções encontradas (intensificação), busca local ou global, entre outros. Estes aspectos são definidos nas próximas subseções.

2.3.0.1 INTENSIFICAÇÃO E DIVERSIFICAÇÃO

Exploração e intensificação são duas características comuns e fundamentais de qualquer método de otimização. No entanto, é altamente dependente da maneira de busca adotada por cada metaheurística (BLUM et al., 2012). Esses dois fatores são considerados essenciais

para a solução adequada de um problema de otimização. A exploração é a capacidade de expandir a pesquisa em um amplo domínio de soluções, com o objetivo de explorar áreas não visitadas do espaço de busca. Por sua vez, a intensificação utiliza a experiência de busca armazenada, permitindo focar regiões promissoras (soluções de alta qualidade) para convergir de maneira ideal (BOUSSAÏD et al., 2013).

Para equilibrar esses dois fatores, um algoritmo metaheurístico eficiente deve procurar ou explorar novas soluções, através de técnicas de randomização e buscas aleatórias, para que o movimento exploratório acesse todas as regiões do espaço de busca por pelo menos uma vez (YANG, 2014).

Resumidamente, no processo de intensificação as regiões promissoras do espaço de busca são exploradas de maneira minuciosa, com o objetivo de encontrar as melhores soluções. Já na diversificação ou exploração, regiões não exploradas devem ser visitadas para assegurar que todas as regiões do espaço de busca sejam exploradas equilibradamente e que a busca não fique limitada a um pequeno espaço de regiões.

2.3.1 TÉCNICAS PARA APRIMORAMENTO DA METAHEURÍSTICA PSO

Várias técnicas estão sendo empregadas para o aprimoramento da metaheurística PSO tradicional com os objetivos principais de melhoria entre as capacidades de exploração e intensificação do algoritmo, escape dos pontos mínimos locais e maior rapidez de convergência. Dentre essas técnicas, este trabalho utiliza a distribuição Beta, mapas caóticos e a sequência de Halton para aprimorar a qualidade das soluções do PSO tradicional, sendo descritas nas próximas seções.

2.3.1.1 DISTRIBUIÇÃO BETA

A função de Distribuição Beta é um tipo de distribuição probabilística contínua, definida no intervalo parametrizado $[0, 1]$ por dois parâmetros positivos $p > 0$ e $q > 0$, os quais são expoentes da variável aleatória e controlam o formato da distribuição (MACA; PECH, 2015). Essa distribuição tem sido aplicada para modelar o comportamento de variáveis aleatórias limitadas a intervalos de tamanho finito em uma vasta quantidade de problemas. A distribuição Beta é um modelo adequado para o comportamento aleatório de porcentagens e proporções.

A função de distribuição Beta para vários valores dos parâmetros p e q é ilustrada na

Figura 3 e definida por:

$$p = \frac{\mu^2(1 - \mu)}{\sigma^2} - 1 \quad (2)$$

$$q = (1 - \mu) \left[\frac{\mu(1 - \mu)}{\sigma^2} - 1 \right] \quad (3)$$

sendo μ a média das amostras e σ^2 a variância das amostras x .

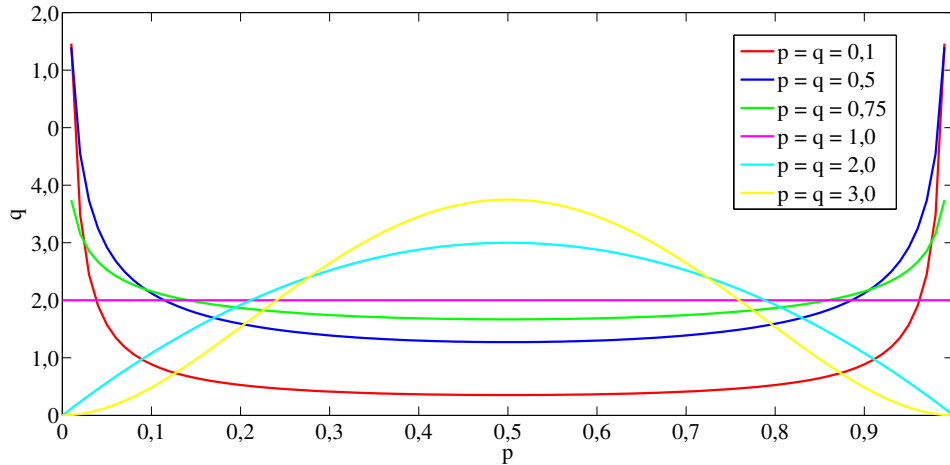


Figura 3: Distribuição Beta com vários valores de p e q .

Fonte: Autoria própria

A função de densidade de probabilidade (PDF) da distribuição Beta, para $0 \leq x \leq 1$, e os parâmetros de forma $p, q > 0$, é uma função de potência da variável x e de seu reflexo $(1 - x)$, respectivamente. da seguinte forma:

$$\begin{aligned} f(x; p, q) &= \kappa \cdot x^{p-1}(1 - x)^{q-1} \\ &= \frac{1}{\text{Beta}(p, q)} \cdot x^{p-1}(1 - x)^{q-1} \end{aligned} \quad (5)$$

sendo a função $\text{Beta}(p, q)$ uma constante de normalização tal que

$$\kappa = \frac{1}{\text{Beta}(p, q)} = \frac{\Gamma(p + q)}{\Gamma(p) \cdot \Gamma(q)} = \int_0^1 u^{p-1}(1 - u)^{q-1} du$$

garante que a probabilidade total seja 1, sendo $\Gamma(\cdot)$ a função Gama. Assim, nas equações acima, x é uma realização (uma amostra, um valor observado) do processo aleatório (p.a.) $X \sim \text{Beta}(p, q)$.

2.3.1.2 MAPAS CAÓTICOS

Um dos maiores desafios das metaheurísticas é encontrar o equilíbrio apropriado entre diversificação e intensificação (BONYADI; MICHALEWICZ, 2017). Nesse contexto, a aplicação de mapas caóticos tem sido proposta para melhorar a qualidade das soluções encontradas pela metaheurística PSO tradicional, considerando sua capacidade de escape de pontos mínimos locais no processo de otimização (ALATAS et al., 2009); (YANG et al., 2014); (ARAUJO; COELHO, 2008); (PLUHACEK et al., 2015); (GAO et al., 2012). Os mapas caóticos incluem uma grande variedade de esquemas baseados no comportamento complexo e caótico de sistemas não-lineares, possuindo natureza não-repetitiva, o que aumenta as características de busca do algoritmo (YANG et al., 2014). A Figura 4 ilustra os diversos tipos de mapas caóticos que podem ser utilizados para a melhoria do processo de busca do algoritmo PSO.

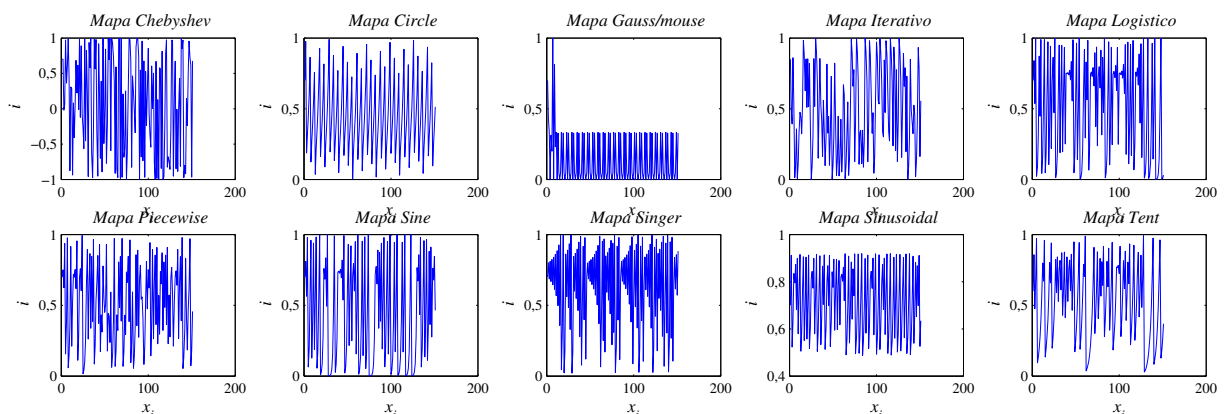


Figura 4: Tipos de Mapas Caóticos.

2.3.1.3 SEQUÊNCIA DE HALTON

Vários tipos de métodos Monte Carlo são empregados para a geração de números aleatórios, dentre eles, tem-se o método Quasi-Monte Carlo que produz números quase randômicos altamente uniformes (CHI; JONES, 2006). Esses métodos são amplamente empregados em várias áreas, especialmente em problemas com domínios multidimensionais (DIGEHSARA et al., 2020) (CHI; JONES, 2006). A sequência de Halton é um tipo de método Quasi-Monte Carlo, utilizada em muitas aplicações por produzir sequências com baixa discrepância (DIGEHSARA et al., 2020). A sequência de Halton emprega números aleatórios com estrutura matemática que fortalece a fase de exploração das metaheurísticas, provendo cobertura apropriada do espaço de busca (DIGEHSARA et al., 2020). Dessa maneira, essa sequência pode auxiliar a metaheurística PSO na fase de geração da população inicial do exame, provendo a inicialização mais apropriada das posições das partículas. A Figura 5(a)

e 5(b) ilustram o efeito da função randômica e a sequência de Halton para a geração da população inicial, respectivamente, com 500 partículas em um espaço de busca de duas dimensões. Como pode-se perceber, a Figura 5(a) possui várias áreas do espaço de busca que não foram cobertas pelas partículas, enquanto essas mesmas áreas são uniformemente cobertas pela sequência de Halton, como mostrado na Figura 5(b). Pode-se notar que os pontos gerados pela sequência de Halton são mais uniformes.

Formalmente, o i -ésimo elemento na sequência de Halton baseado no número primo p é obtido através do radical inverso do inteiro i na base b , sendo dado por (CHI; JONES, 2006):

$$\varphi_p(i) = \sum_{l=0}^L b_l(i) p^{-l-1} \quad (6)$$

cujos os valores de $\{b_0(i), \dots, b_L(i)\}$ podem ser definidos a partir de:

$$i = \sum_{l=0}^L b_l(i) p^l \quad (7)$$

sendo $0 \leq b_l(i) \leq p-1$ e $p^l \leq i \leq p^{l+1}$.

A maior *desvantagem* das sequências de Halton está relacionada à utilização de números primos muito grandes, pois nesse caso a sequência pode apresentar alta correlação e, conseqüentemente, inapropriada cobertura do espaço de busca e baixa qualidade e desempenho no processo de simulação e otimização dos algoritmos metaheurísticos (CHI; JONES, 2006). Para contornar esse problema é utilizada a sequência de Halton Embaralhada, com o objetivo de improvisar o comportamento da sequência de Halton original, através da redução da correlação em altas dimensões ao mesmo tempo em que mantém boa qualidade para problemas com apenas uma dimensão (STEINERBERGER, 2020). Esse método embaralha as sequências de 1-D utilizando permutações dos coeficientes $b_l(i)$ no radical inverso da função. Assim, o i -ésimo elemento da sequência Halton Embaralhada para um determinado número primo p é dado por:

$$\varphi_{sp}(i) = \sum_{l=0}^L \Pi_p(b_l(i)) p^{-l-1} \quad (8)$$

Π_p é o operador de permutação para os possíveis valores de $b_l(i)$ na base p . Geralmente, os números inteiros $b_l(i)$ são primos e maiores que 1. No Capítulo 4 é demonstrado o processo de escolha das variáveis $b_l(i)$ e Π_p realizado por este trabalho.

Vários métodos são utilizados para produzir as permutações Π_p , dentre eles, o mais popular é o método de Braaten Weller (B&W) (TUFFIN, 1998). Esse método fornece diferentes permutações para diferentes valores de subconjuntos de números primos σ_p . Depois do

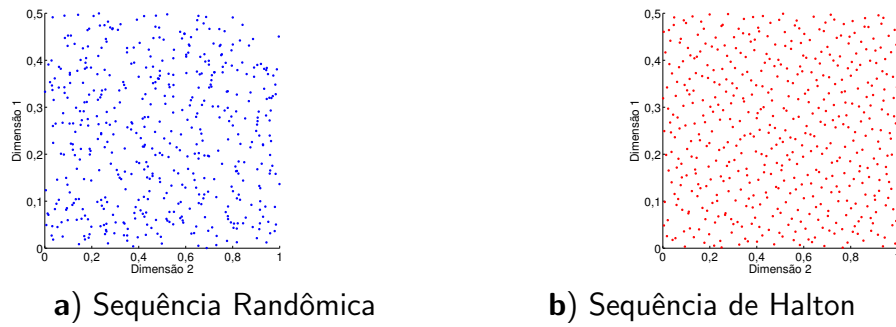


Figura 5: 500 partículas iniciais para o PSO com (a) Sequência Randômica (b) Sequência de Halton.

Fonte: Autoria própria

embaralhamento, o resultado da sequência não apresenta forte correlação entre as dimensões do problema (STEINERBERGER, 2020). Os passos básicos do método B&W podem ser resumidos em:

- ◇ Inicializar: $\sigma_{r0} = 0$;
- ◇ Conjunto $b = [1, \dots, b - 1]$;
- ◇ Repetir $r = 1$ até $b - 1$;
- ◇ Escolher σ_r do Conjunto para minimizar a discrepância;
- ◇ Conjunto = Conjunto / σ_r ;

O procedimento de permutações da sequência (Π_S) proposto em (TUFFIN, 1998) para os primeiros 16 números primos é mostrado na Tabela 1. Para ilustrar o procedimento de embaralhamento, é apresentado um exemplo da permutação de σ_p números primos como $\Pi_{\sigma_p=3} = (0\ 1\ 2)$ para o conjunto de números primos σ_3 . Neste método, o quinto número na base 3 da sequência de Halton na forma digitalizada é 0,21, que quando expandido na base 3 corresponde a $1 \times 3^{-1} + 2 \times 3^{-2} = 5/9$. Dessa forma, os primeiros 8 números na sequência embaralhada correspondente à base 3 são definidos por: 2/3, 1/3, 2/9, 8/9, 5/9, 1/9, 7/9, 4/9. Isso é ilustrado na Figura 6(a), que mostra 1000 pontos da sequência de Halton embaralhada, correspondentes a duas dimensões. Considerando que o valor base depende da dimensão fornecida, se a dimensão for d , as bases serão os primeiros d números primos (ou seja, para a dimensão $d = 5$, as bases são [2, 3, 5, 7, 11]). Para fins de comparação, as sequências de Halton padrão são representadas na Figura 6(b), na qual percebe-se claramente que não houve cobertura mais uniforme do espaço de busca.

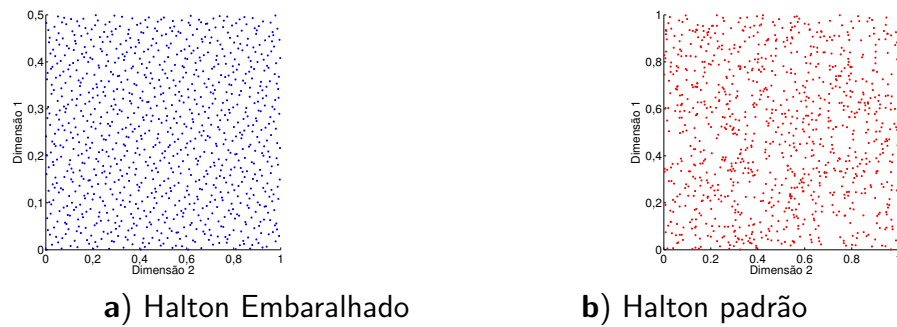


Figura 6: 1000 pontos gerados por (a) Sequência de Halton Embaralhada; (b) Sequência de Halton.

Fonte: Autoria própria

Tabela 1: Permutações dos 16 primeiros números primos obtidos pelo método B&W.

Conjunto # Primos	Sequência da Permutações (Π_S)
σ_2	(0 1)
σ_3	(0 2 1)
σ_5	(0 2 4 1 3)
σ_7	(0 3 5 1 6 2 4)
σ_{11}	(0 5 8 2 10 3 6 1 9 4 7)
σ_{13}	(0 6 10 2 8 4 12 1 9 5 11 3 7)
σ_{17}	(0 8 13 3 11 5 16 1 10 7 14 4 12 2 15 6 9)
σ_{19}	(0 9 14 3 17 6 11 1 15 7 12 4 18 8 2 16 10 5 13)
σ_{23}	(0 11 17 4 20 7 13 2 22 9 15 5 18 1 14 10 21 6 16 3 19 8 12)
σ_{29}	(0 14 22 5 18 9 27 2 20 11 25 7 16 3 24 13 19 6 28 10 1 23 15 12 26 4 17 8 21)
σ_{31}	(0 16 8 26 4 22 13 29 2 19 11 24 6 20 14 28 1 17 9 30 10 23 5 21 15 3 27 12 25 7 18)
σ_{37}	(0 18 28 6 23 11 34 3 25 14 31 8 20 36 1 16 27 10 22 13 32 4 29 17 7 35 19 2 26 12 30 9 24 15 33 5 21) 21)
σ_{41}	(0 20 31 7 26 12 38 3 23 34 14 17 29 5 40 10 24 1 35 18 28 9 33 15 21 4 37 13 30 8 39 19 25 2 32 11 22 36 6 27 16)
σ_{43}	(0 21 32 7 38 13 25 3 35 17 28 10 41 5 23 30 15 37 1 19 33 11 26 42 8 18 29 4 39 14 22 34 6 24 12 40 2 31 20 16 36 9 27)
σ_{47}	(0 23 35 8 41 14 27 3 44 18 31 11 37 5 25 39 16 21 33 1 46 12 29 19 42 7 28 10 36 22 4 43 17 32 13 38 2 26 45 15 30 6 34 20 40 9 24)
σ_{53}	(0 26 40 9 33 16 49 4 36 21 45 12 29 6 51 23 38 14 43 1 30 19 47 10 34 24 42 3 27 52 15 18 39 7 46 22 32 5 48 13 35 25 8 44 31 17 50 2 37 20 28 11 41)

2.4 CALIBRAÇÃO DOS PARÂMETROS

Para a calibração dos parâmetros de entrada do algoritmo PSO é utilizada a técnica Golden-Section (GS) (CHANG, 2009), que consiste em um método numérico de otimização muito similar ao procedimento sistemático proposto por (FILHO et al., 2014). Nesse método, os parâmetros principais de entrada que afetam drasticamente o desempenho dos algoritmos são otimizados, como por exemplo, os coeficientes de aprendizado cognitivo (c_1) e social (c_2). Após, os parâmetros de entrada relacionados a complexidade do algoritmo são otimizados, considerando o compromisso entre complexidade e desempenho, como por exemplo, o número de iterações $\mathcal{I}$ e a quantidade de partículas $\mathcal{S}$. Para o exemplo dos parâmetros citados, a técnica GS consiste em duas etapas:

- Manter c_1 fixo e otimizar c_2 ;
- c_2 (otimizado na etapa anterior) é mantido fixo enquanto o valor de c_1 é otimizado.

Os valores dos parâmetros de entrada otimizados para o algoritmo PSO utilizado nesta Tese são encontrados pelo método GS, que otimiza o mínimo de uma função custo ao estreitar sucessivamente a faixa de valores dentro da faixa viável; ou seja, estima os valores máximo e mínimo do parâmetro de entrada até que o melhor valor seja encontrado para o mesmo. Os processos de calibração dos parâmetros de entrada dos algoritmos Chaos-PSO, β -PSO e Halton-PSO são explicados nos próximos capítulos.

2.5 ANÁLISE DAS METAHEURÍSTICAS

Nesta seção, são apresentados os aspectos relevantes sobre a implementação das metaheurísticas utilizadas neste trabalho para a solução de problemas de otimização, assim como os critérios adotados para convergência e complexidade computacional.

2.5.1 CRITÉRIO DE PARADA

O critério de parada ideal para um algoritmo metaheurístico é aquele que garante que todos os parâmetros convirjam independentemente (RENNER; EKÁRT, 2003). No entanto, isso pode ser muito exigente ou resultar em muitas iterações; portanto, critérios de convergência mais flexíveis são geralmente empregados. Neste trabalho, para todas as metaheurísticas é utilizado o mesmo critério de parada, considerando:

- Número máximo de iterações: usado para evitar que o algoritmo gaste muito tempo refinando uma solução existente;
- Erro quadrático médio normalizado (NMSE): utilizado para medir a qualidade dos algoritmos.

Assim, é realizado um teste de convergência ($\mathcal{C}$), considerando que após o resultado da i -ésima iteração, y^i é factível e o algoritmo finaliza sua execução se obtiver 100% de sucesso na seguinte relação (9):

$$\mathcal{C} = \|y^i - y^{i-1}\| < \xi = 10^{-2} \quad (9)$$

A factibilidade dos algoritmos metaheurísticos depende do problema de otimização considerado. Geralmente, os métodos de otimização consideram que o valor nulo da factibilidade indica que o valor otimizado satisfaz as restrições do problema (BOUSSAÏD et al., 2013). A qualidade da solução alcançada pelas metaheurísticas em uma iteração pode ser medida pelo grau de proximidade em relação à solução ótima $\mathbf{o}^*$, sendo quantificada através do erro quadrático médio normalizado (NMSE - Normalized Mean Square Error) quando o equilíbrio é atingido. A qualidade da solução alcançada por um algoritmo específico na solução do problema na i -ésima iteração é dada por (10):

$$\text{NMSE}[t, i] = \mathbb{E} \left[\frac{\|\mathbf{o}_t[i] - \mathbf{o}^*\|^2}{\|\mathbf{o}^*\|^2} \right] \quad (10)$$

sendo $\|\cdot\|^2$ a norma Euclidiana ao quadrado entre o vetor $\mathbf{o}_t[i]$ para o vetor da solução ótima $\mathbf{o}^*$ na i -ésima iteração da t -ésima realização do processo de otimização; $\mathbb{E}[\cdot]$ denota o operador esperança estatística.

2.5.2 ROBUSTEZ E COMPLEXIDADE COMPUTACIONAL

A robustez $\mathcal{R}$ das metaheurísticas apresentadas neste trabalho é definida por:

$$\mathcal{R} = \frac{\text{Sc}}{\text{Trials}} \cdot 100 \quad [\%] \quad (11)$$

no qual Sc representa o número de sucessos de convergência e Trials denota o número total de realizações dos algoritmos. Para avaliar a complexidade das metaheurísticas utilizadas foi considerado o número de operações por ponto flutuante (FLOPs), sendo as mesmas: adição, subtração, multiplicação e divisão (WU; ZHU, 2020). O número de FLOPs para cada algoritmo metaheurístico foi obtido através da utilização do programa CountFlops (QIAN, 2015).

2.6 FUNÇÕES BECHMARK

Na matemática aplicada, as funções de testes, também conhecidas como Benchmark, são úteis para avaliar características das metaheurísticas de otimização, como: taxa de convergência, precisão, robustez e desempenho. Nesta seção, são apresentadas as funções Benchmark, ilustradas em duas dimensões, que são utilizadas para validar a metaheurística proposta por este trabalho.

2.6.1 FUNÇÃO ACKLEY

A função Ackley é não-convexa, não-separável e não-diferenciável, sendo caracterizada por uma região externa quase plana e uma grande abertura no centro. A função representa um risco para algoritmos de otimização, particularmente por ficarem presos em um de seus muitos mínimos locais. Essa função pode ser definida por qualquer entrada, mas geralmente é avaliada no domínio $x_i \in [-32, 32]$, para todo $i = 1, \dots, d$, no qual d é a dimensão do problema. O mínimo global dessa função é definido em eq. (12) e ilustrado na Figura 7 para $d = 2$:

$$f(\mathbf{x}^*) = 0, \quad \text{em} \quad \mathbf{x}^* = (0, \dots, 0) \quad (12)$$

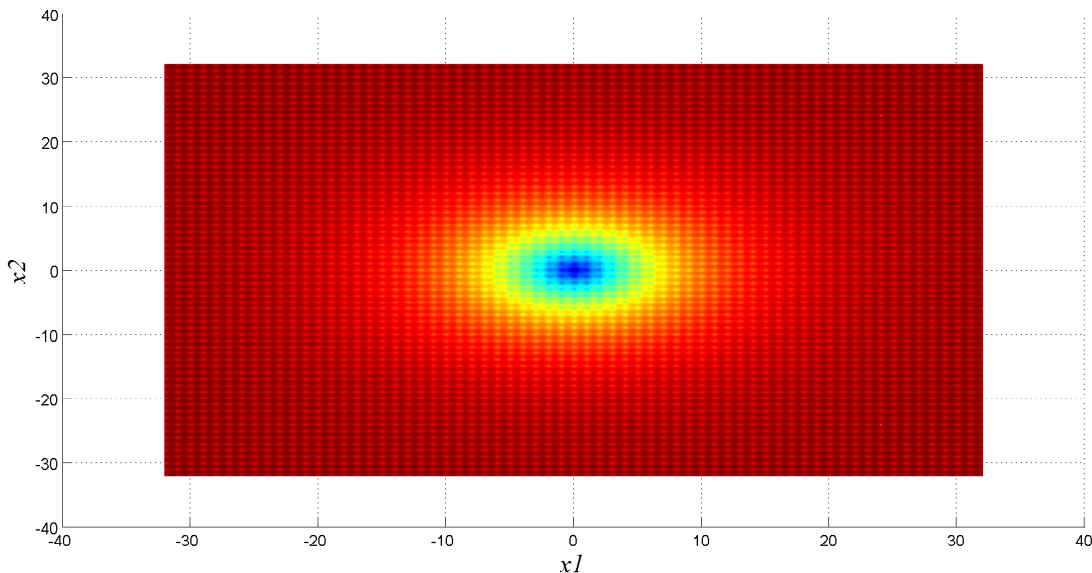


Figura 7: Mínimo Global da Função Ackley de $d = 2$ dimensões.

Fonte: Autoria própria

A função Ackley é definida pela equação (13) e ilustrada em duas dimensões na Figura

8.

$$f(\mathbf{x}) = -a \cdot \exp \left(-b \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(cx_i) \right) + a + \exp(1) \quad (13)$$

Os valores recomendados para os parâmetros livres são: $a = 20$, $b = \frac{1}{5}$ e $c = 2\pi$.

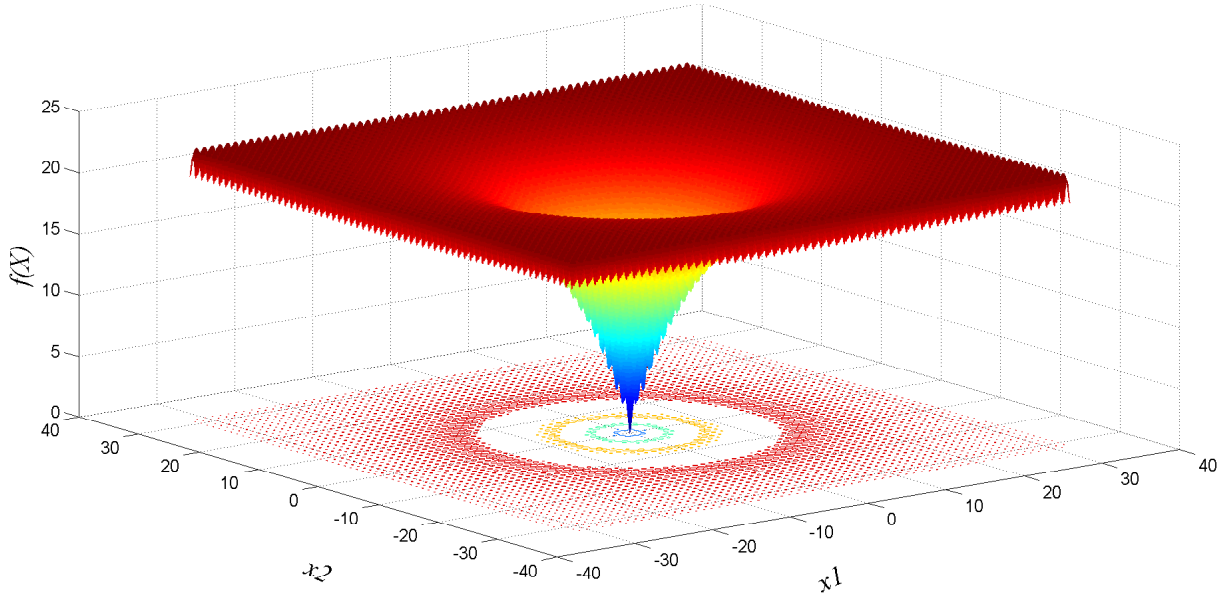


Figura 8: Função de teste Ackley.

Fonte: Autoria própria

2.6.2 FUNÇÃO GRIEWANK

A função Griewank possui vários mínimos locais, que são regularmente distribuídos. Essa função geralmente é avaliada no domínio $x_i \in [-600, 600]$, para todo $i = 1, \dots, d$ e seu mínimo global é dado por:

$$f(\mathbf{x}^*) = 0, \quad \text{em} \quad \mathbf{x}^* = (0, \dots, 0) \quad (14)$$

A função de teste Griewank é expressa de acordo com (15) e ilustrada na Figura 9.

$$f(\mathbf{x}) = f(x_1, \dots, x_n) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) \quad (15)$$

Já a Figura 10 apresenta, para melhor visualização dos pontos de mínimos locais, a função Griewank ampliada no intervalo $[-20; 20]$; as respectivas curvas de nível para o mesmo intervalo são apresentadas na Figura 11.

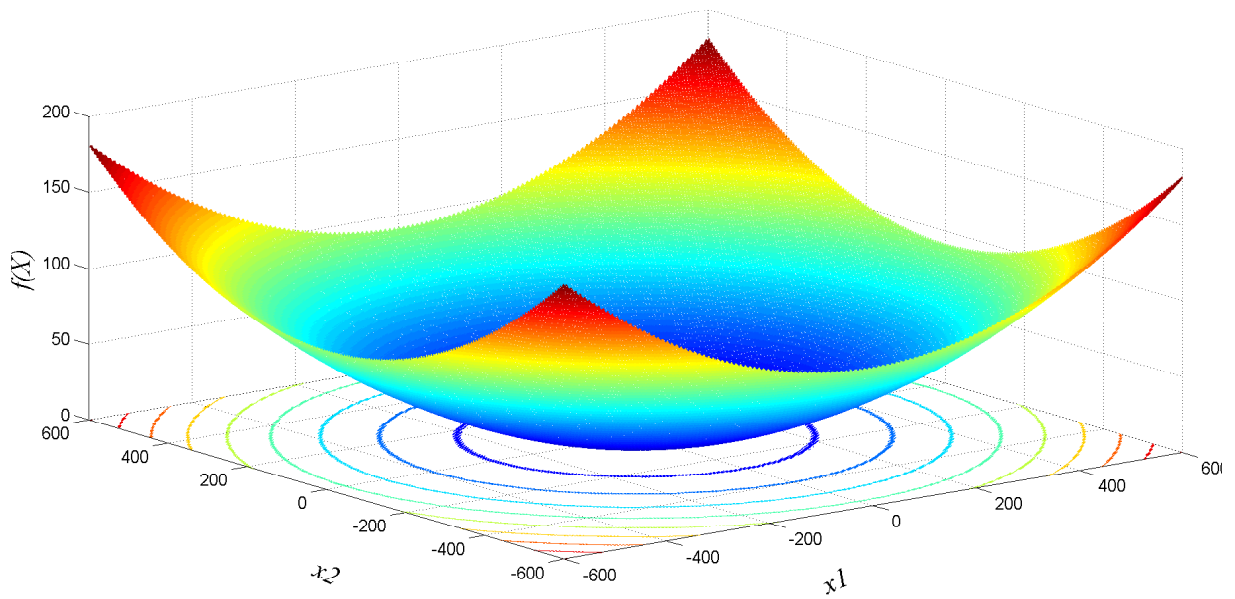


Figura 9: Função de Teste Griewank.

Fonte: Autoria própria

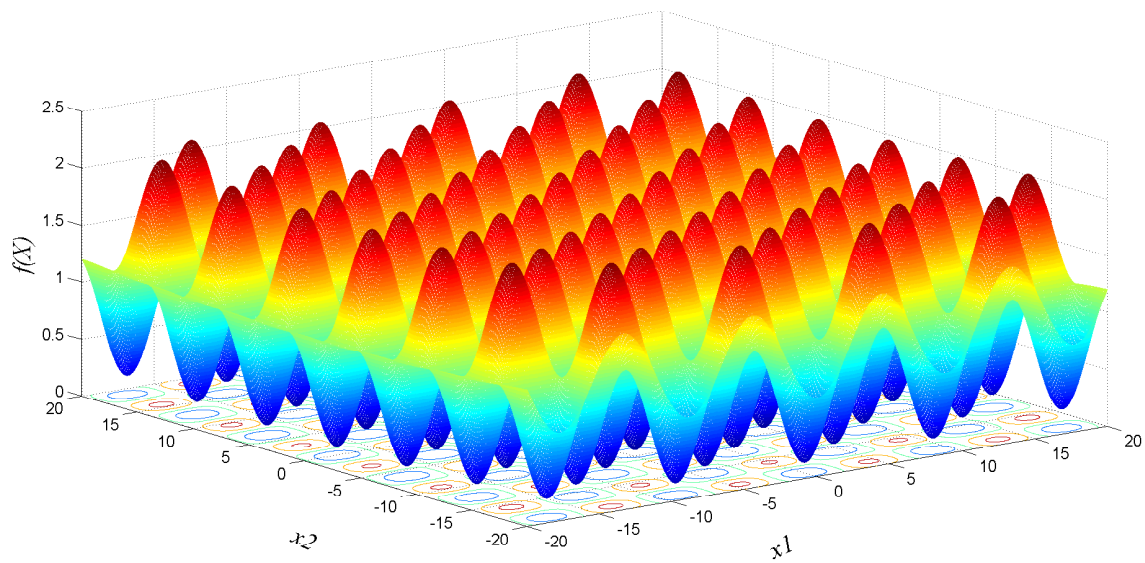


Figura 10: Função Griewank ampliada.

Fonte: Autoria própria

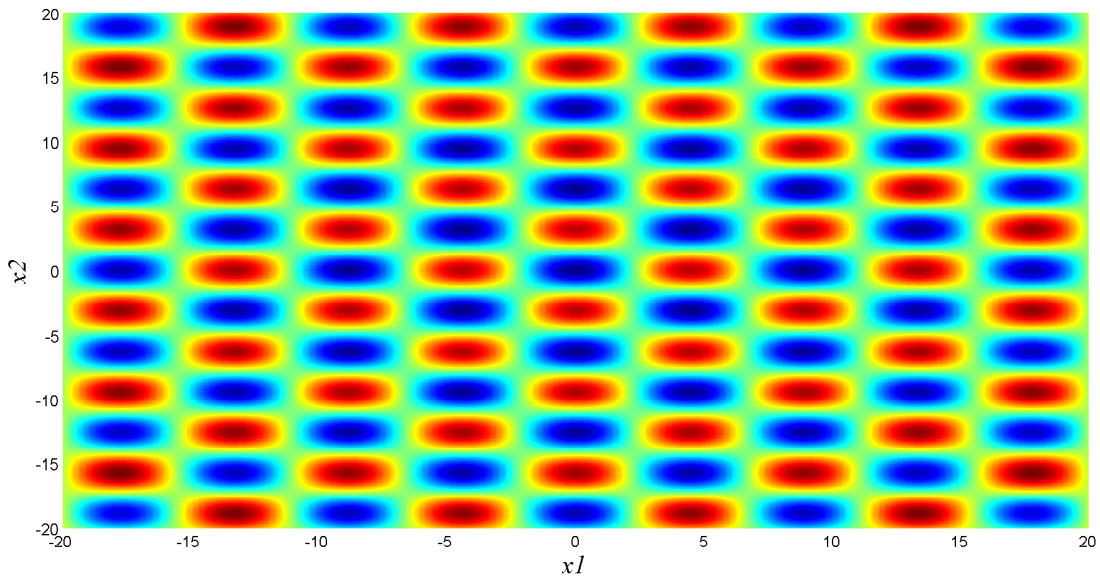


Figura 11: Mínimos Globais da Função Griewank.

Fonte: Autoria própria

2.6.3 FUNÇÃO ROSENBROCK

A função Rosenbrock, também conhecida como função Valley, é um problema de teste popular para algoritmos de otimização. Essa função é não-convexa e seu mínimo global fica localizado em um vale estreito e parabólico. Embora esse vale seja fácil de encontrar, a convergência para este ponto mínimo é difícil de ser alcançada (Picheny et al., 2012). A função pode ser definida em qualquer domínio de entrada, mas geralmente é avaliada em $x_i \in [-5, 10]$, para todo $i = 1, \dots, d$, sendo ilustrada na Figura 12 para $d = 2$ dimensões, cujo mínimo global é ilustrado na Figura 12.(b), sendo definido por:

$$f(\mathbf{x}^*) = 0, \quad \text{em } \mathbf{x}^* = (1, \dots, 1), \quad \text{para } a = 1. \quad (16)$$

A função de teste Rosenbrock é definida de acordo com:

$$f(\mathbf{x}) = \sum_{i=1}^{d-1} [b(x_{i+1} - x_i^2)^2 + (a - x_i)^2], \quad (17)$$

na qual as constantes geralmente apresentam os valores (típicos) definidos por $a = 1$ e $b = 100$.

2.6.4 FUNÇÃO RASTRINGIN

A função de teste Rasingin é convexa, altamente multimodal, não-linear e possui vários pontos de mínimos locais, que são regularmente distribuídos. Essa função pode ser

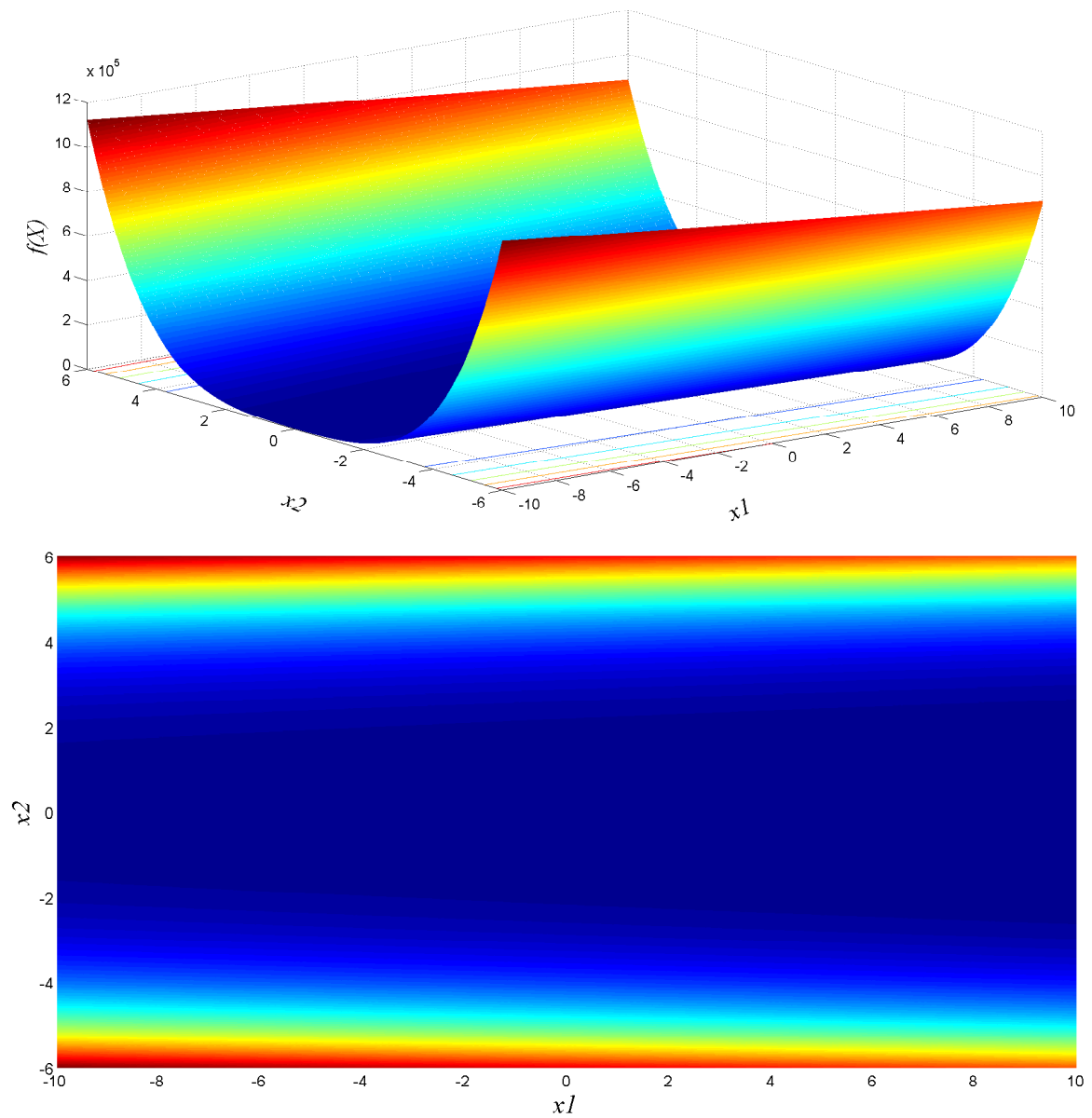


Figura 12: Função de teste Rosenbrock e respectivo mínimo global.

Fonte: Autoria própria

definida em qualquer domínio, mas usualmente é avaliada em $x_i \in [-5, 12, 5, 12]$, para todo $i = 1, \dots, d$. Seu mínimo global é apresentado na Figura 13, sendo definido por:

$$f(\mathbf{x}^*) = 0, \quad \text{em} \quad \mathbf{x}^* = (0, \dots, 0) \quad (18)$$

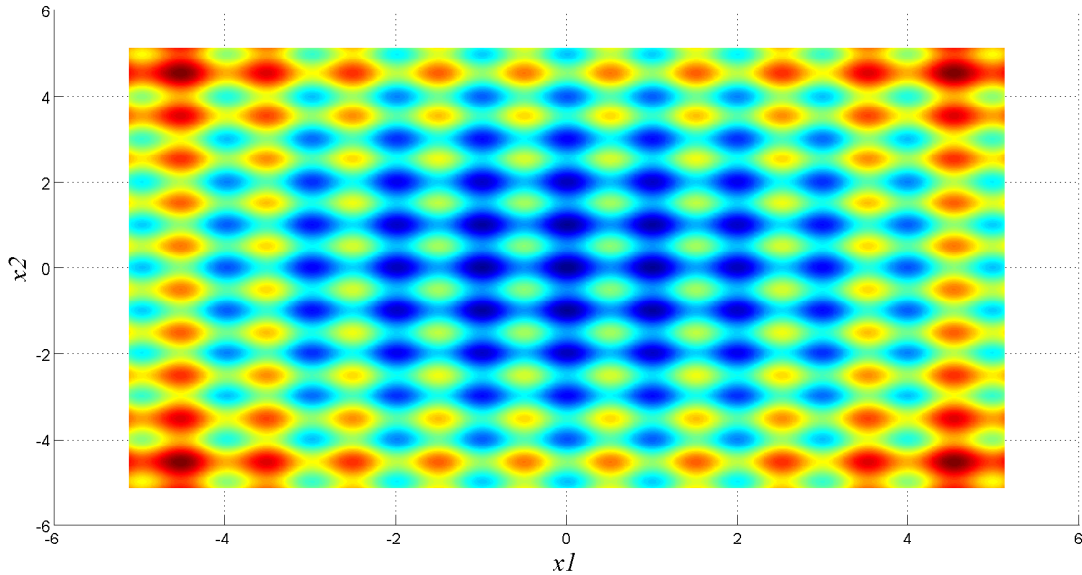


Figura 13: Mínimo Global da Função Rastrigin.

Fonte: Autoria própria

A função Rastrigin de dimensão d é definida por:

$$f(\mathbf{x}) = f(x_1, \dots, x_d) = 10d + \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i)), \quad (19)$$

sendo ilustrada na Figura 14 para $d = 2$ dimensões.

2.7 CONCLUSÕES DO CAPÍTULO

Neste capítulo, foram apresentados os conceitos teóricos fundamentais sobre otimização e metaheurísticas bioinspiradas. Além disso, foram apresentadas as técnicas de calibração de parâmetros utilizadas neste trabalho, assim como alguns critérios de convergência, complexidade computacional e robustez que são adotados para a validação e avaliação da metaheurística proposta. Por fim, foram apresentadas as funções Benchmark que servirão de base para a validação do algoritmo proposto Halton-Chaos- β -PSO.

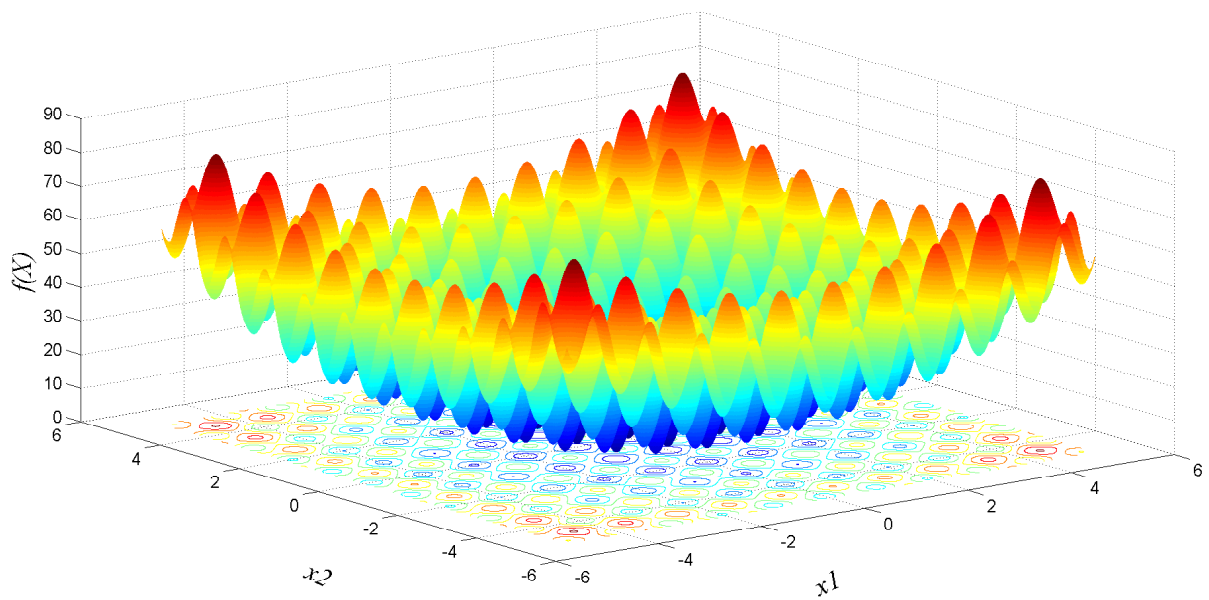


Figura 14: Função de teste Rastrigin.

Fonte: Autoria própria

3 METAHEURÍSTICAS UTILIZADAS

Neste Capítulo, são apresentadas as metaheurísticas utilizadas por este trabalho, especialmente a metaheurística proposta Halton-Chaos- β -PSO, assim como é realizada a validação desses algoritmos utilizando funções Benchmark e comparando seu compromisso entre desempenho e complexidade com o PSO tradicional.

3.1 METAHEURÍSTICA DE OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

A Otimização por enxame de Partículas é uma metaheurística populacional e estocástica utilizada para a otimização de problemas através de uma busca iterativa por um valor máximo ou mínimo baseado em uma função custo (EBERHART; KENNEDY, 1995). Esse algoritmo busca por uma solução candidata dentre uma população, chamada de partículas, usando parâmetros matemáticos que determinam a posição e a velocidade das mesmas (YANG, 2014).

O PSO é inicializado com um enxame aleatório de partículas, as quais representam potenciais soluções para um problema. O movimento de cada partícula, ilustrado na Figura 15, depende da melhor posição individual encontrada até o momento e da melhor posição encontrada pelo enxame no espaço de busca. Conforme novas e promissoras posições são encontradas, essas informações são utilizadas para guiar o movimento das partículas pelo espaço de busca. No PSO, a população inicial é definida por:

$$\mathcal{S} = \{\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_p; \dots; \mathbf{x}_P\} \quad (20)$$

sendo P o número de partículas ou o tamanho do enxame. Ademais, a posição de cada partícula no PSO é definida por:

$$x_i = [x_{p_1} \ x_{p_2} \ \dots \ x_{p_j}]^T \in \mathcal{A}; \quad j = 1, 2, \dots, \mathcal{I}_{\max} \quad (21)$$

onde $\mathcal{I}_{\max}$ é o número máximo de iterações.

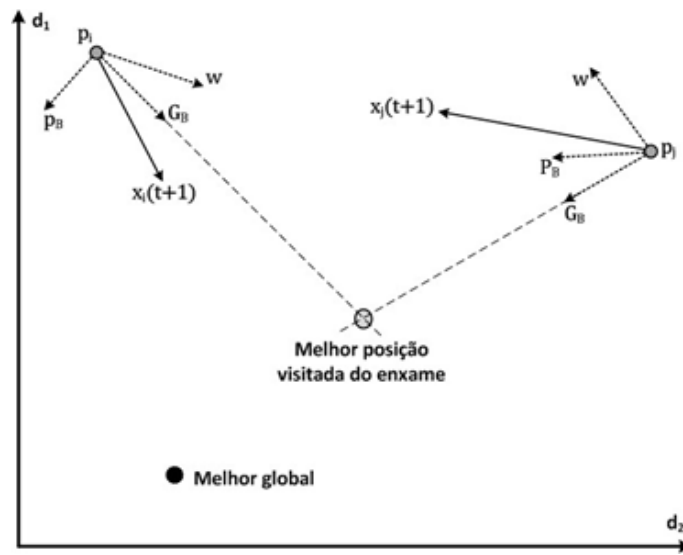


Figura 15: Deslocamento das partículas no algoritmo PSO Tradicional.

Fonte: Adaptado de (EBERHART; KENNEDY, 1995)

Considere x_i^j a posição da i -ésima partícula na j -ésima iteração no espaço de busca, então a posição dessa partícula na próxima iteração (24) é atualizada através da adição da velocidade v_i^{j+1} para a posição atual, conforme (23):

$$v_i^{j+1} = \omega \cdot v_i^j + c_1 \cdot \mathcal{U}_1(p_i^* - x_i^j) + c_2 \cdot \mathcal{U}_2(g_i^* - x_i^j) \quad (22)$$

$$x_i^{j+1} = x_i^j + v_i^{j+1} \quad (23)$$

onde ω é o peso da inércia, x_i^j e v_i^j denotam a posição e a velocidade da partícula i na iteração j , respectivamente; c_1 e c_2 representam as constantes de aceleração usadas para agregar a contribuição dos aspectos cognitivo e social, respectivamente; g_i^* e p_i^* denotam o melhor valor global e individual obtidos até o momento; $\mathcal{U}_1$ e $\mathcal{U}_2$ são os valores das variáveis aleatórias obtidas através da distribuição uniforme no intervalo $[0, 1]$.

Basicamente, o algoritmo PSO consiste em três etapas: a) avaliação de cada partícula; b) atualização do melhor valor individual e global das soluções candidatas; c) atualização da velocidade e posição de cada partícula. Esse processo se repete de maneira iterativa com o objetivo (mesmo que não garantido) de encontrar uma solução suficientemente satisfatória, por exemplo, soluções quase ótimas. O algoritmo PSO tradicional é mostrado no Algoritmo 1.

O PSO é uma das metaheurísticas bioinspiradas mais simples de ser implementada, porém apresenta como maior desvantagem a convergência prematura para mínimos ou máximos locais. Para contornar esse problema, são agregadas algumas técnicas, como:

Algoritmo 1: Algoritmo PSO Tradicional

```

1  início
2  | Inicializar: população com distribuição uniforme;
3  repita
4  |   para  $i = 1$  até  $P$  faça
5  |       se  $f(g_i^*) < f(x_i)$  então
6  |           |  $x_i = g_i^*$ ;
7  |       fim
8  |       Avaliar função custo;
9  |       para  $j = 1$  até  $\mathcal{I}_{\max}$  faça
10 |           | Atualizar velocidade usando eq. 22;
11 |           | Atualizar posição usando eq. 23;
12 |       fim
13 |   fim
14 até Critério de parada (9) é satisfeito;
15 fim

```

fixação de velocidade máxima para as partículas, peso variável da inércia (SHI; EBERHART, 1998); (SENGUPTA et al., 2019), fator de constrição (CLERC, 1999), operador de vizinhança (SUGANTHAN, 1999), entre outras. Neste trabalho, ao PSO tradicional são agregadas as seguintes modificações, discutidas nas próximas seções: Distribuição Beta para geração de variáveis aleatórias, Mapa Caótico como fator de amortecimento e sequência de Halton Embaralhada a geração da população inicial do algoritmo.

3.2 METAHEURÍSTICA β -PSO

A metaheurística β -PSO utilizada neste trabalho é baseada no algoritmo PSO tradicional, mas utiliza a distribuição Beta, discutida na seção 2.3.1.1, ao invés da distribuição uniforme para a geração das variáveis aleatórias, com o objetivo de aumentar a diversificação e ajudar na exploração de regiões encobertas do espaço de busca durante o processo de otimização (SANTANA et al., 2018). A distribuição Beta foi previamente empregada para melhorar a metaheurística PSO em relação à atualização do peso inercial (MACA; PECH, 2015) e inicialização da velocidade das partículas (ANDRÉ et al., 2015).

Neste trabalho, a distribuição Beta é utilizada para a geração das variáveis aleatórias utilizadas para a atualização da velocidade das partículas, considerando o domínio $[0, 1]$ para diferentes valores dos parâmetros p e q . Nesse contexto, a velocidade das partículas da metaheurística β -PSO é calculada por (24):

$$v_i^{j+1} = \omega \cdot v_i^j + c_1 \cdot \mathcal{B}_1(p_i^* - x_i^j) + c_2 \cdot \mathcal{B}_2(g_i^* - x_i^j) \quad (24)$$

onde $\mathcal{B}_1$ e $\mathcal{B}_2$ representam matrizes diagonais, nas quais seus elementos são valores aleatórios geradas pela distribuição Beta no intervalo $\mathcal{B}(p, q) \in [0, 1]$ gerado para a i -ésima partícula na iteração $j = 1, 2, \dots, \mathcal{P}$. O Algoritmo 2 descreve o funcionamento dessa metaheurística.

Algoritmo 2: Algoritmo β -PSO

```

1  início
2  | Inicializar: população com distribuição uniforme;
3  | repita
4  |   para  $i = 1$  até  $P$  faça
5  |   | se  $f(g_i^*) < f(x_i)$  então
6  |   | |  $x_i = g_i^*$ ;
7  |   | fim
8  |   | Avaliar função custo;
9  |   | para  $j = 1$  até  $\mathcal{I}_{\max}$  faça
10 |   | | Atualizar velocidade usando eq. 24;
11 |   | | Atualizar posição usando eq. 23;
12 |   | fim
13 |   fim
14 | até Critério de parada (9) é satisfeito;
15 fim

```

3.3 METAHEURÍSTICA CHAOS-PSO

A metaheurística Chaos-PSO (C-PSO) é baseada no algoritmo PSO tradicional e utiliza mapas caóticos principalmente para a geração de variáveis aleatórias necessárias no processo de inicialização ou atualização da velocidade ou posição das partículas do enxame. Alguns exemplos de mapas caóticos agregados ao algoritmo PSO para a geração de variáveis aleatórias podem ser encontrados em (YANG et al., 2014). Adicionalmente, alguns algoritmos C-PSO utilizam mapas caóticos para encontrar novas soluções na vizinhança das melhores posições já encontradas pelo algoritmo (ALATAS et al., 2009); (YANG et al., 2014).

Neste trabalho, um fator de amortecimento baseado em mapas caóticos é utilizado para a atualização do melhor valor global encontrado pelo algoritmo. Esse aspecto pode evitar a convergência prematura, limitando a dominância do melhor valor global e aumentando a aleatoriedade (diversificação) sem perda na capacidade de exploração (SANTANA et al., 2018). Essa modificação pode afetar positivamente o compromisso entre desempenho e com-

plexidade do algoritmo. As características fundamentais desse fator de amortecimento podem ser ilustradas por dois importantes casos assintóticos de acordo com:

$$e^{-L(g^*[t]-x_i[t])} \rightarrow 0 \quad \text{e} \quad e^{-L(g^*[t]-x_i[t])} \rightarrow 1 \quad (25)$$

Para $e^{-L(g^*[t]-x_i[t])} \rightarrow 0$, a diferença entre o melhor valor global e a partícula candidata ($g^*[t]-x_i[t]$) deve ser alta, isto é, o melhor global e a partícula candidata estão muito distantes no espaço de busca; assim, a influência do melhor valor global é diminuída. Por outro lado, quando $e^{-L(g^*[t]-x_i[t])} \rightarrow 1$, o melhor valor global e a partícula candidata estão bem próximas. Nesse caso, a influência do melhor valor global é aumentada sobre a partícula candidata.

Considerando essas características, a atualização da velocidade para o algoritmo C-PSO é dada por (26):

$$v_i^{j+1} = \omega \cdot v_i^j + c_1 \cdot \mathcal{U}_1(p_i^* - x_i^j) + c_2 \cdot e^{L(g_i^* - x_i^j)} \cdot \mathcal{U}_2(g_i^* - x_i^j) \quad (26)$$

onde $L \in [0, 1]$ é o fator de amortecimento gerado pelo mapa caótico logístico unidimensional, que está relacionado à dinâmica da população biológica (??). O mapa logístico é definido de acordo com:

$$L[j+1] = \xi \cdot L[j](1 - L[j]) \quad (27)$$

onde ξ é o parâmetro de controle do fator de amortecimento. A variação de $L[j]$ pode aumentar a aleatoriedade da influência do melhor valor global. A atualização da posição das partículas é calculada da mesma maneira que o PSO tradicional e o pseudocódigo do algoritmo C-PSO pode ser visualizado no Algoritmo 3.

3.4 HALTON-PSO

A inicialização da população do PSO afeta a convergência global e a eficiência da capacidade de busca desse algoritmo (YANG, 2014), sendo que a adequada definição do conjunto da população inicial pode influenciar na qualidade das soluções obtidas (EBERHART; KENNEDY, 1995); (YANG, 2014). Tradicionalmente, o algoritmo PSO inicializa a posição das partículas de maneira uniforme e aleatória. Por outro lado, a metaheurística Halton-PSO (H-PSO) (DIGEHSARA et al., 2020) utiliza sequências de Halton como mecanismo para a geração da população inicial na primeira etapa do processo de otimização, objetivando

Algoritmo 3: Algoritmo C-PSO

```

1  início
2  Inicializar: população com distribuição uniforme;
3  repita
4      para  $i = 1$  até  $P$  faça
5          se  $f(g_i^*) < f(x_i)$  então
6               $x_i = g_i^*$ ;
7          fim
8          Avaliar função custo;
9          para  $j = 1$  até  $\mathcal{I}_{\max}$  faça
10             Atualizar velocidade usando eq. 26;
11             Atualizar posição usando eq. 23;
12         fim
13     fim
14 até Critério de parada (9) é satisfeito;
15 fim
  
```

apropriada cobertura do espaço viável de busca. Através da utilização da sequência de Halton Embaralhada, responsável pela geração de números quase aleatórios, a população inicial do algoritmo é gerada de maneira mais eficiente e sem afetar diretamente a estrutura do algoritmo PSO original.

Neste trabalho, a sequência de Halton Embaralhada é utilizada com o objetivo de produzir diversidade nas soluções iniciais e aumentar a capacidade de exploração do algoritmo nos passos iniciais do processo de otimização (DIGEHSARA et al., 2020). O processo de atualização da velocidade e posição das partículas é realizado da mesma maneira que o PSO tradicional, sendo o pseudocódigo do algoritmo H-PSO descrito pelo Algoritmo 4.

3.5 HALTON-CHAOS- β -PSO

Este trabalho propõe o desenvolvimento de uma nova abordagem metaheurística para otimização, chamada Halton-Chaos- β -PSO (HC- β -PSO), que é baseada em três características específicas que foram agregadas ao PSO tradicional:

- a) Distribuição Beta: esta distribuição é adotada ao invés da distribuição uniforme para a geração de variáveis aleatórias com o objetivo de aumentar a diversidade e auxiliar na exploração (diversificação) de regiões encobertas no espaço de busca durante o processo de otimização;
- b) Fator de amortecimento: este fator é baseado em números aleatórios gerados por mapas

Algoritmo 4: Algoritmo H-PSO

```

1  início
2  Inicializar: população com sequência de Halton Embaralhada;
3  repita
4      para  $i = 1$  até  $P$  faça
5          se  $f(g_i^*) < f(x_i)$  então
6               $x_i = g_i^*$ ;
7          fim
8          Avaliar função custo;
9          para  $j = 1$  até  $\mathcal{I}_{\max}$  faça
10             Atualizar velocidade usando eq. 22;
11             Atualizar posição usando eq. 23;
12         fim
13     fim
14 até Critério de parada (9) é satisfeito;
15 fim
  
```

caóticos e utilizado para a atualização do melhor valor global. Essa característica pode limitar a dominância do melhor valor global e evitar convergência prematura, aumentando a aleatoriedade (diversificação) sem ocasionar perda na capacidade de exploração do algoritmo;

c) Sequência de Halton: esta sequência é empregada para melhorar a qualidade de geração da população inicial no primeiro passo do algoritmo com o objetivo de melhorar a cobertura do espaço viável de busca.

Considerando tais características agregadas, a atualização da velocidade da metaheurística HC- β -PSO é realizada de acordo com:

$$v_i^{j+1} = \omega \cdot v_i^j + c_1 \cdot \mathcal{B}_1(p_i^* - x_i^j) + c_2 \cdot e^{L(g_i^* - x_i^j)} \cdot \mathcal{B}_2(g_i^* - x_i^j) \quad (28)$$

da mesma maneira que é definido no algoritmo β -PSO, $\mathcal{B}_1$ e $\mathcal{B}_2$ são matrizes diagonais com elementos aleatórios definidos pela distribuição Beta no intervalo $\mathcal{B}(p, q) \in [0, 1]$, gerados para a i -ésima partícula na iteração $j = 1, 2, \dots, \mathcal{P}$. Nota-se que na equação de atualização da velocidade do HC- β -PSO é utilizado um fator de amortecimento pertencente ao algoritmo C-PSO, na equação (26), além disso os parâmetros da distribuição Beta da metaheurística β -PSO são agregados. Como resultado, a metaheurística proposta HC- β -PSO para problemas de otimização é descrita no pseudocódigo a seguir (5).

Como principais vantagens do algoritmo metaheurístico proposto pode-se citar o aumento da diversidade das partículas candidatas, melhorando a diversificação, ou seja, conse-

Algoritmo 5: Algoritmo HC- β -PSO

```

1  início
2  Inicializar: população com sequência de Halton Embaralhada;
3  repita
4      para  $i = 1$  até  $P$  faça
5          se  $f(g_i^*) < f(x_i)$  então
6               $x_i = g_i^*$ ;
7          fim
8          Avaliar função custo;
9          para  $j = 1$  to  $\mathcal{I}_{\max}$  faça
10             Atualizar velocidade usando eq. 28;
11             Atualizar posição usando eq. 23;
12         fim
13     fim
14 até Critério de parada (9) é satisfeito;
15 fim
  
```

guindo identificar regiões encobertas no espaço de busca durante o processo de otimização devido à distribuição Beta. Além disso, o uso de um fator de amortecimento baseado em mapas caóticos pode limitar a dominância do melhor valor global, evitando convergência prematura e aumentando a diversidade, sem ocasionar perda na capacidade de exploração do algoritmo. Adicionalmente, a utilização da sequência de Halton Embaralhada para gerar a população inicial proporciona melhor cobertura do espaço viável de busca do algoritmo. Tais recursos ajudam a melhorar a velocidade de convergência e a precisão das soluções, contribuindo para a redução da estimativa de erro e, consequentemente, melhoria da qualidade dos resultados, como pode ser observado na Seção 3.6 e nos Capítulos 5 e 6.

3.6 RESULTADOS NUMÉRICOS DAS METAHEURÍSTICAS VALIDADAS COM FUNÇÕES BENCHMARK

Nesta seção, as metaheurísticas PSO tradicional, β -PSO, C-PSO e HC- β -PSO são validadas e comparadas através das funções Benchmark apresentadas na seção 2.6. Para todas as metaheurísticas foi realizada a calibração dos parâmetros de entrada, assim como o cálculo da complexidade computacional, expresso em termos de FLOPs, considerando as dimensões $d = [2, 5, 10, 20, 50]$ para todas as funções de teste utilizadas. Como as metaheurísticas β -PSO, C-PSO e HC- β -PSO possuem qualidade de resultado similar das soluções do PSO tradicional em termos de compromisso entre desempenho-complexidade para os mesmos parâmetros de entrada, esses valores são utilizados para todas as metaheurísticas. Por outro lado, para a metaheurística β -PSO são ajustados os parâmetros relacionados à distribuição Beta, en-

quanto para o C-PSO e H-PSO, o fator de amortecimento e o coeficiente de combinação dos números primos são calibrados, respectivamente. A Tabela 2 mostra os valores dos domínios das funções Benchmark utilizados nesta Seção, sendo que para a função Ackley foram utilizados os valores 20, 0,2 e 2π para os parâmetros livres l_1 , l_2 e l_3 , respectivamente. Já para a função Rosenbrock, os valores das constantes foram definidos em $l_4 = 1$ e $l_5 = 100$.

Tabela 2: Valores dos domínios das Funções Benchmark.

Função Benchmark	Domínio
Ackley	$D = [-32, 32]$
Griewank	$D = [-600, 600]$
Rosenbrock	$D = [-5, 10]$
Rastrigin	$D = [-5, 12, 5, 12]$
Esfera	$D = [-5, 12, 5, 12]$

3.6.1 CALIBRAÇÃO DOS PARÂMETROS DAS METAHEURÍSTICAS PARA AS FUNÇÕES BENCHMARK

Nas próximas seções, são apresentados os resultados das calibrações dos parâmetros de entrada de todas as metaheurísticas para as funções Benchmark utilizadas neste trabalho. Para a calibração dos parâmetros, foram adotados os seguintes passos:

- Calibração dos parâmetros c_1 , c_2 e ω da metaheurística PSO tradicional;
- Calibração dos parâmetros $\mathcal{S}$ e $\mathcal{I}_{\max}$ da metaheurística PSO tradicional;
- Calibração dos parâmetros p e q da metaheurística β -PSO;
- Calibração do parâmetro ξ da metaheurística C-PSO;
- Calibração do parâmetro b da metaheurística H-PSO.

3.6.1.1 CALIBRAÇÃO DOS PARÂMETROS DA METAHEURÍSTICA PSO TRADICIONAL

As Tabelas 3, 4, 5 e 6 mostram os valores dos parâmetros do PSO tradicional calibrados para as funções Benchmark, considerando diferentes dimensões para as mesmas. O intervalo de valores adotado foi entre $[-4, 5]$ para os parâmetros c_1 , c_2 e ω . Vale ressaltar que esses valores também são utilizados para as metaheurísticas baseadas no PSO.

Tabela 3: Parâmetros da Metaheurística PSO Tradicional para a função Ackley.

Parâmetros	Dimensões				
	2	5	10	20	50
Aceleração da partícula (c_1)	1,5	1,2	-0,3	0,1	-0,2
Aceleração global (c_2)	1,5	1,3	3,8	3,3	3,7
Peso da inércia (ω)	0,4	0,8	-0,4	-0,6	-0,2
Número de partículas ($\mathcal{S}$)	25	30	52	60	100
Máximo de iterações ($\mathcal{I}_{\max}$)	20	50	180	300	300

Tabela 4: Parâmetros da Metaheurística PSO Tradicional para a função Griewank.

Parâmetros	Dimensões				
	2	5	10	20	50
Aceleração da partícula (c_1)	1,5	1,5	1,8	-0,5	-0,2
Aceleração global (c_2)	1,5	2,0	2,0	3,3	3,9
Peso da inércia (ω)	0,5	0,6	0,7	-0,3	-0,5
Número de partículas ($\mathcal{S}$)	25	30	30	70	100
Máximo de iterações ($\mathcal{I}_{\max}$)	20	50	120	220	300

Tabela 5: Parâmetros da Metaheurística PSO Tradicional para a função Rastrigin.

Parâmetros	Dimensões				
	2	5	10	20	50
Aceleração da partícula (c_1)	2,0	1,8	1,8	-1,0	-0,2
Aceleração global (c_2)	2,0	2,2	2,0	3,8	3,7
Peso da inércia (ω)	0,5	0,3	0,7	-0,5	-0,2
Número de partículas ($\mathcal{S}$)	25	60	80	60	100
Máximo de iterações ($\mathcal{I}_{\max}$)	15	100	200	270	300

Tabela 6: Parâmetros da Metaheurística PSO Tradicional para a função Rosenbrock.

Parâmetros	Dimensões				
	2	5	10	20	50
Aceleração da partícula (c_1)	2,5	2,6	-0,3	-1,0	-0,2
Aceleração global (c_2)	1,3	1,0	4,9	3,8	3,9
Peso da inércia (ω)	0,4	0,5	-0,3	-0,5	-0,3
Número de partículas ($\mathcal{S}$)	25	202	52	60	105
Máximo de iterações ($\mathcal{I}_{\max}$)	20	100	100	200	300

3.6.1.2 CALIBRAÇÃO DOS PARÂMETROS DA METAHEURÍSTICA β -PSO

Para a metaheurística β -PSO foram calibrados os parâmetros p e q referentes à distribuição Beta, sendo adotado o intervalo de valores $p \in [0, 1; 4, 0]$. Conforme a metodologia adotada por este trabalho, o melhor valor encontrado para o parâmetro p , ou seja, o valor que possui o menor número de iterações para a convergência do algoritmo, é fixado para então se encontrar o melhor valor para o parâmetro q , com o objetivo de definir o melhor par de valores (p, q) para cada dimensão das funções. Após a execução de vários testes, os valores dos parâmetros p e q para as funções Benchmark foram fixados, considerando a avaliação de uma vasta quantidade de combinações dos parâmetros, mas para fins práticos, apenas os valores mais representativos foram adotados no decorrer deste trabalho.

A Figura 16 ilustra o número de iterações necessárias para a convergência do algoritmo β -PSO para a função Ackley com diferentes dimensões, empregando diferentes valores para o parâmetro p . Nota-se que o valor desse parâmetro pode influenciar na velocidade de convergência do algoritmo, já que a variação do número necessário de iterações oscila entre $\mathcal{I} \approx 10$ ($p = 1,5$) e $\mathcal{I} \approx 20$ ($p = 0,5$) para 2 dimensões, de acordo com a Figura 16(a). Já nas Figuras 16(b) e 16(c) o número de iterações para a convergência variou entre $\mathcal{I} \approx 20$ ($p = 1,4$) e $\mathcal{I} \approx 35$ ($p = 2,8$), e $\mathcal{I} \approx 85$ ($p = 2,0$) e $\mathcal{I} \approx 150$ ($p = 1,1$) para 5 e 10 dimensões, respectivamente. As Figuras 16(d) e 16(e) ilustram o número de iterações necessárias para a convergência considerando 20 e 50 dimensões da função Ackley, respectivamente, e pode-se observar que este número oscilou entre $\mathcal{I} \approx 170$ ($p = 2,7$) e $\mathcal{I} \approx 210$ ($p = 2,3$), e $\mathcal{I} \approx 220$ ($p = 3,8$) e $\mathcal{I} \approx 200$ ($p = 1,5$).

Na Figura 17, é ilustrado o número de iterações necessárias para a convergência do algoritmo β -PSO para a função Griewank com diferentes dimensões, empregando diferentes valores para o parâmetro p . Assim como na função apresentada anteriormente, pode-se observar a oscilação do número de iterações em relação ao parâmetro p . Para a função Griewank com 2 e 5 dimensões, apresentadas em 17(a) e 17(b), nota-se que quando o valor de $(p = 1,1)$ e $(p = 1,3)$, o número de iterações exigidas é $\mathcal{I} \approx 10$ e $\mathcal{I} \approx 32$, respectivamente, sendo menor que em outros casos. Na Figura 17(c), tem-se os valores de p para a função com 10 dimensões e nota-se que $\mathcal{I} \approx 90$ ($p = 1,2$) e $\mathcal{I} \approx 110$ ($p = 1,4$). Analisando a influência do parâmetro p em relação à velocidade de convergência são também apresentados os cenários para 20 e 50 dimensões nas Figuras 17(d) e 17(e), respectivamente; para 20 dimensões $\mathcal{I} \approx 170$ ($p = 1,5$) e $\mathcal{I} \approx 220$ ($p = 1,0$); enquanto que para 50 dimensões a oscilação foi $\mathcal{I} \approx 190$ ($p = 2,5$) e $\mathcal{I} \approx 240$ ($p = 1,7$).

Para a função Rastrigin também são analisados cenários com diferentes dimensões e

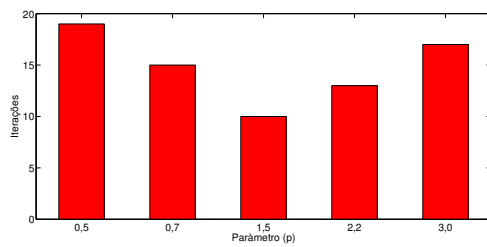
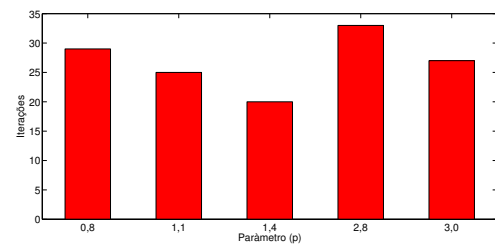
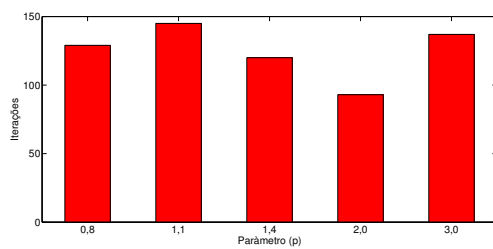
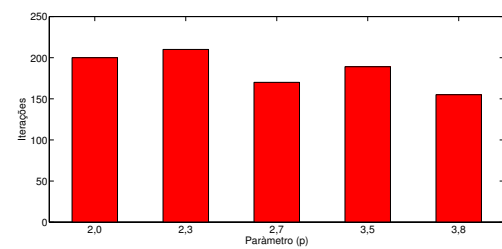
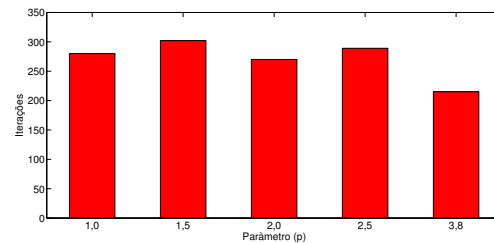
a) Parâmetro p para Função Ackley 2-Db) Parâmetro p para Função Ackley 5-Dc) Parâmetro p para Função Ackley 10-Dd) Parâmetro p para Função Ackley 20-De) Parâmetro p para Função Ackley 50-D

Figura 16: Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Ackley considerando diferentes valores para o parâmetro p .

Fonte: Autoria própria

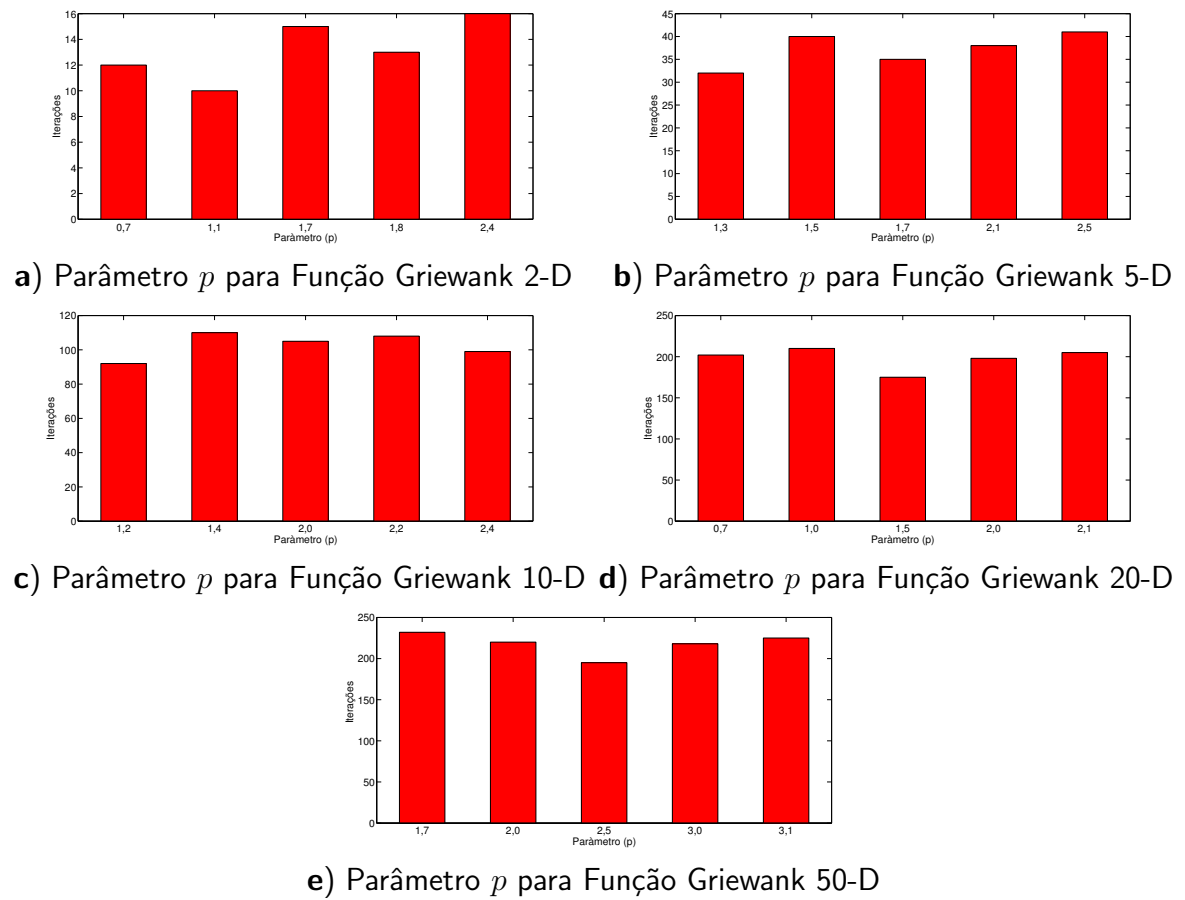
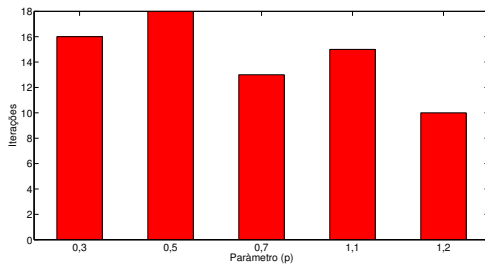


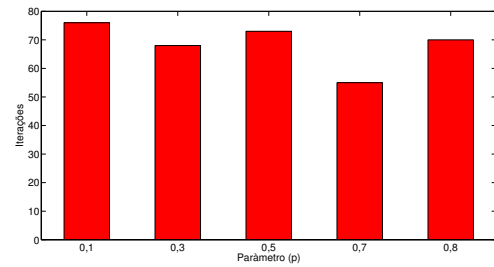
Figura 17: Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Griewank considerando diferentes valores para o parâmetro p .

Fonte: Autoria própria

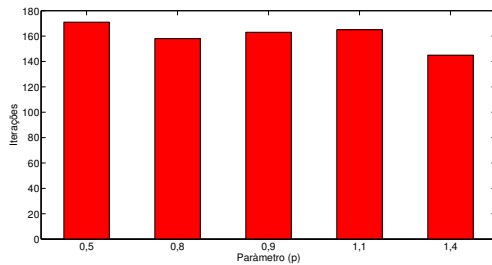
iterrações necessrias em relao  variao do parmetro p , que pode ser observado na Figura 18. Para o cenrio com 2 dimenses,  apresentada a Figura 18(a) e nota-se que oscilao entre o nmero de iterraes necessrias para a convergncia variou entre $\mathcal{I} \approx 10$ ($p = 1,2$) e $\mathcal{I} \approx 18$ ($p = 0,5$). J para 5 dimenses, a oscilao ficou entre $\mathcal{I} \approx 55$ ($p = 0,7$) e $\mathcal{I} \approx 75$ ($p = 0,1$), de acordo com a Figura 18(b). Considerando os cenrios para 10 e 20 dimenses, so apresentadas as Figuras 18(c) e 18(d), respectivamente. Nota-se que para 10 dimenses, $\mathcal{I} \approx 150$ ($p = 1,4$) e $\mathcal{I} \approx 170$ ($p = 0,5$), enquanto que com 20 dimenses $\mathcal{I} \approx 190$ ($p = 1,5$) e $\mathcal{I} \approx 230$ ($p = 1,0$). Para o ltimo cenrio, tem-se a Figura 18(e) que ilustra que o melhor desempenho em relao ao menor nmero de iterraes necessrias ($\mathcal{I} \approx 220$) para 50 dimenses  obtido quando $p = 3,5$.



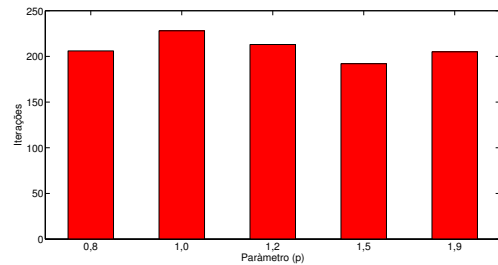
a) Parmetro p para Funo Rastrigin 2-D



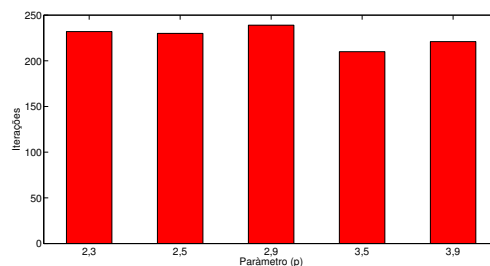
b) Parmetro p para Funo Rastrigin 5-D



c) Parmetro p Funo Rastrigin 10-D



d) Parmetro p Funo Rastrigin 20-D



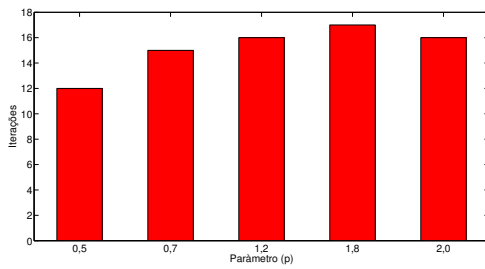
e) Parmetro p para Funo Rastrigin 50-D

Figura 18: Nmero de iterraes necessrias para a convergncia da metaheurstica β -PSO com a funo Rastrigin considerando diferentes valores para o parmetro p .

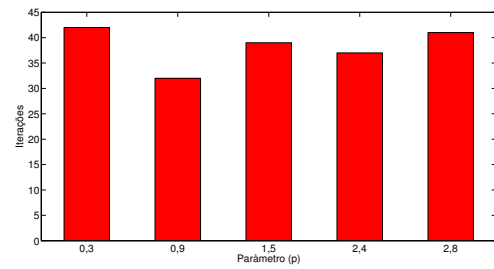
Fonte: Autoria prpria

A Figura 19 ilustra o nmero de iterraes necessrias para a convergncia do algoritmo β -PSO para a funo Rosenbrock com diferentes dimenses e valores para o parmetro

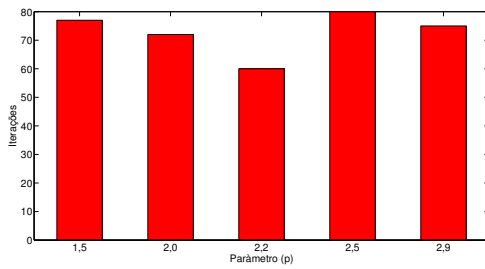
p . Observa-se na Figura 19(a) que a variação do número necessário de iterações para a convergência da função com 2 dimensões variou entre $\mathcal{I} \approx 12$ ($p = 0,5$) e $\mathcal{I} \approx 17$ ($p = 1,8$). Na Figura 19(b), que ilustra a função Rosenbrock com 5 dimensões, percebe-se que a oscilação variou entre $\mathcal{I} \approx 30$ ($p = 0,9$) e $\mathcal{I} \approx 42$ ($p = 0,3$). As Figuras 19(c) e 19(d) ilustram o número de iterações necessárias para a convergência considerando 10 e 20 dimensões, respectivamente, e nota-se a oscilação entre $\mathcal{I} \approx 60$ ($p = 2,2$) e $\mathcal{I} \approx 80$ ($p = 2,5$), e $\mathcal{I} \approx 170$ ($p = 2,9$), e $\mathcal{I} \approx 200$ ($p = 1,5$). Analisando o cenário com 50 dimensões, observa-se na Figura 19(e) que para o valor do parâmetro $p = 3,5$ o número de iterações foi o menor obtido ($\mathcal{I} \approx 180$).



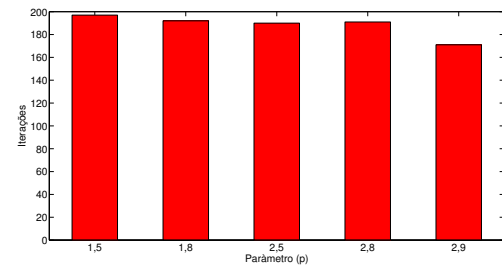
a) Parâmetro p Função Rosenbrock 2-D



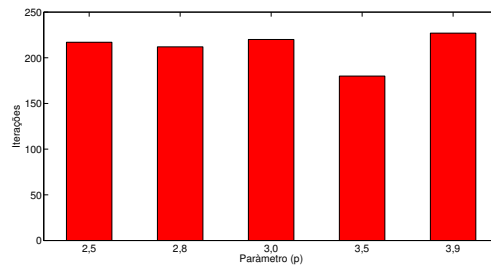
b) Parâmetro p Função Rosenbrock 5-D



c) Parâmetro p Função Rosenbrock 10-D



d) Parâmetro p Função Rosenbrock 20-D



e) Parâmetro p Função Rosenbrock 50-D

Figura 19: Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Rosenbrock considerando diferentes valores para o parâmetro p .

Fonte: Autoria própria

A Tabela 7 mostra os valores de p que obtiveram melhor desempenho, ou seja, aqueles que apresentaram o menor número de iterações necessárias para a convergência considerando cada dimensão das funções.

Tabela 7: Parâmetro p da Metaheurística β -PSO para as funções Benchmark.

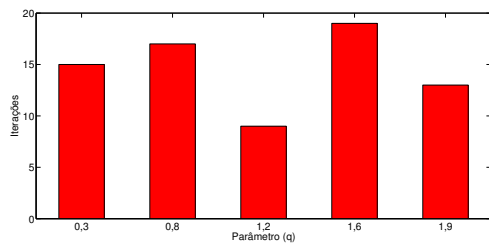
Benchmark	Parâmetro (p)				
	Dimensões				
	2	5	10	20	50
Ackley	1,5	1,4	2,0	2,7	3,8
Griewank	1,1	1,3	1,2	1,5	2,5
Rastrigin	1,2	0,7	1,4	1,5	3,9
Rosenbrock	0,5	0,9	2,2	2,9	3,5

Após a definição dos melhores valores para o parâmetro p , os mesmos foram fixados para a avaliação dos valores para o parâmetro q .

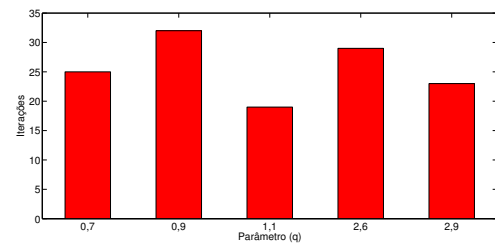
A Figura 20 ilustra o número de iterações que foram necessárias para a convergência do algoritmo β -PSO considerando diferentes valores para o parâmetro q na execução função Ackley. Para o cenário de 2 dimensões, ilustrado na Figura 20(a), foi considerado o parâmetro $p = 1,5$, já que este valor obteve o melhor desempenho, e observa-se que a oscilação para o parâmetro q ficou entre $\mathcal{I} \approx 8$ ($q = 1,2$) e $\mathcal{I} \approx 19$ ($p = 1,6$). Na Figura 20(b), que ilustra a função Ackley com 5 dimensões, percebe-se que a oscilação variou entre $\mathcal{I} \approx 20$ ($q = 1,1$) e $\mathcal{I} \approx 32$ ($p = 0,9$). As Figuras 20(c) e 20(d) ilustram o número de iterações necessárias para a convergência considerando 10 e 20 dimensões, respectivamente, e nota-se a oscilação entre $\mathcal{I} \approx 95$ ($q = 2,2$) e $\mathcal{I} \approx 145$ ($q = 1,7$), e $\mathcal{I} \approx 145$ ($q = 0,7$), e $\mathcal{I} \approx 200$ ($p = 1,1$). Analisando o cenário com 50 dimensões, observa-se na Figura 20(e) que para o valor do parâmetro $q = 0,3$ o número de iterações foi o menor obtido ($\mathcal{I} \approx 205$).

A Figura 21 ilustra o número de iterações necessárias para a convergência do algoritmo β -PSO para a função Griewank com diferentes dimensões, empregando diferentes valores para o parâmetro q . Observa-se que o valor desse parâmetro pode influenciar na velocidade de convergência do algoritmo, já que a variação do número necessário de iterações oscilou entre $\mathcal{I} \approx 10$ ($q = 1,1$) e $\mathcal{I} \approx 18$ ($q = 1,5$) para 2 dimensões, como ilustrado na Figura 21(a). Já nas Figuras 21(b) e 21(c), o número de iterações para a convergência variou entre $\mathcal{I} \approx 30$ ($q = 1,0$) e $\mathcal{I} \approx 38$ ($q = 1,7$) e $\mathcal{I} \approx 95$ ($q = 1,4$) e $\mathcal{I} \approx 120$ ($q = 1,7$) para 5 e 10 dimensões, respectivamente. As Figuras 21(d) e 21(e) ilustram o número de iterações necessárias para a convergência considerando 20 e 50 dimensões da função Griewank, respectivamente, e pode-se observar que este número oscilou entre $\mathcal{I} \approx 180$ ($q = 1,7$) e $\mathcal{I} \approx 200$ ($q = 0,8$; $q = 1,1$) e $\mathcal{I} \approx 190$ ($q = 3,1$), e $\mathcal{I} \approx 220$ ($q = 3,0$).

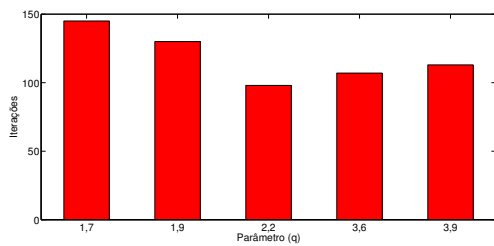
Para a função Rastrigin também foram analisados cenários com diferentes dimensões e iterações necessárias em relação à variação do parâmetro q , que pode ser observado na Figura 22. Para o cenário com 2 dimensões, é apresentada a Figura 22(a) e nota-se que oscilação



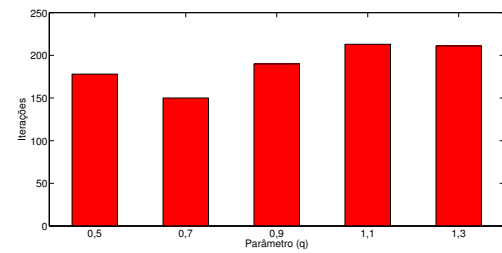
a) Parâmetro q para Função Ackley 2-D



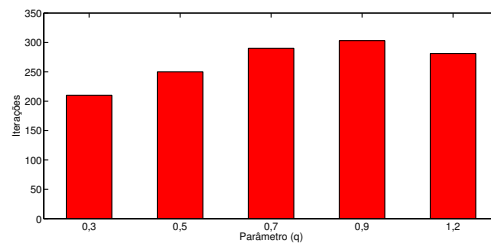
b) Parâmetro q para Função Ackley 5-D



c) Parâmetro q para Função Ackley 10-D



d) Parâmetro q para Função Ackley 20-D



e) Parâmetro q para Função Ackley 50-D

Figura 20: Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Ackley considerando diferentes valores para o parâmetro q .

Fonte: Autoria própria

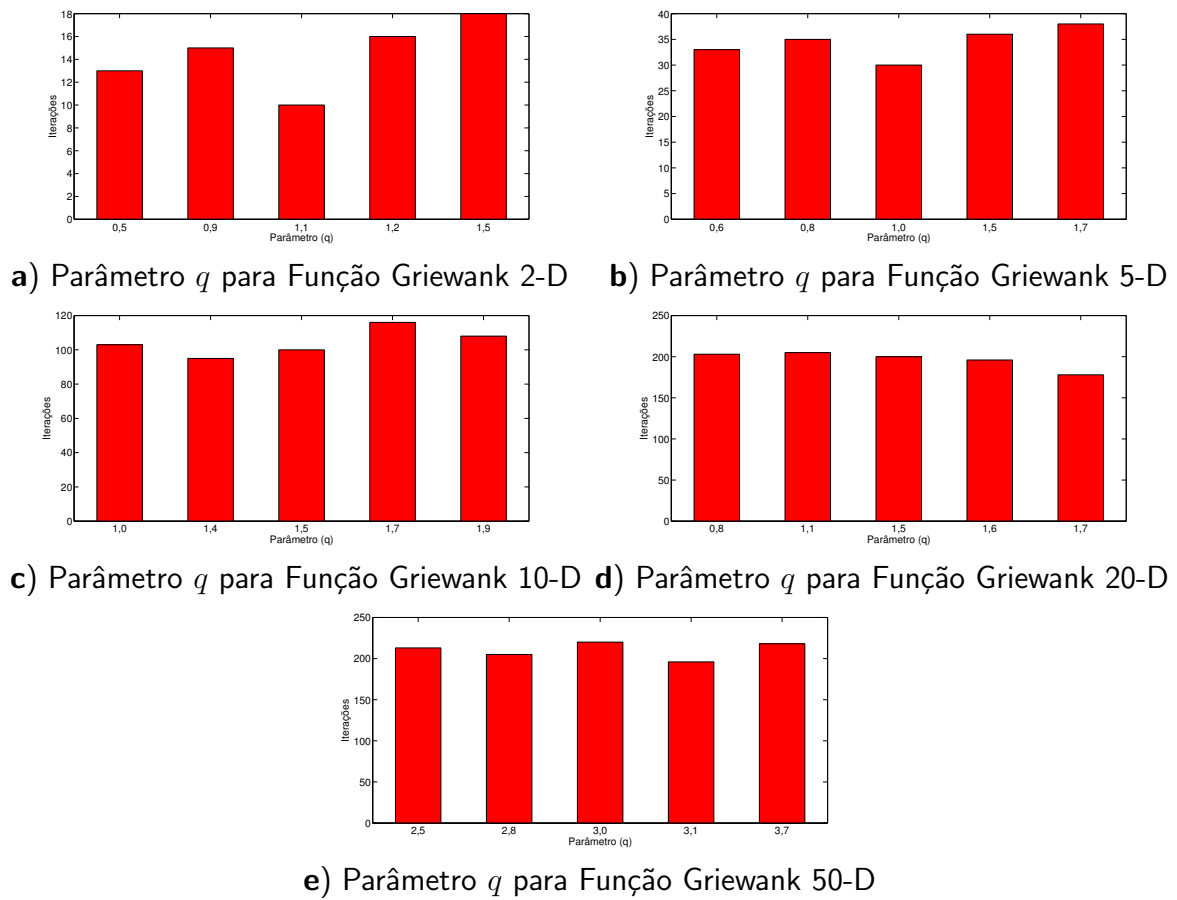
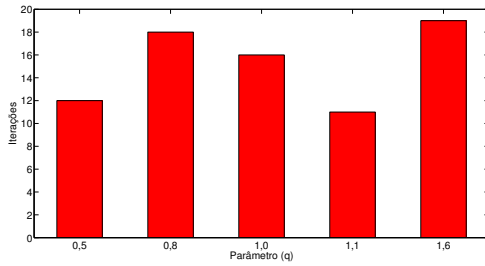


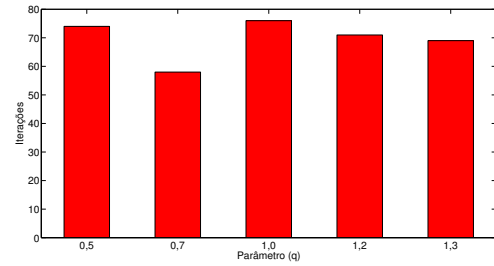
Figura 21: Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Griewank considerando diferentes valores para o parâmetro q .

Fonte: Autoria própria

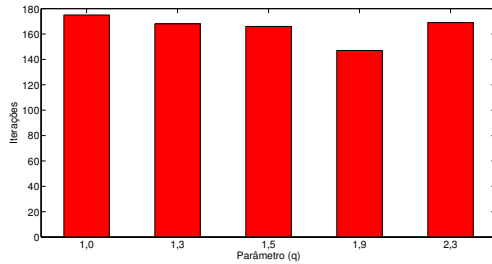
entre o número de iterações necessária para a convergência variou entre $\mathcal{I} \approx 10$ ($p = 1,1$) e $\mathcal{I} \approx 19$ ($q = 1,6$). Analisando para 5 dimensões, a oscilação ficou entre $\mathcal{I} \approx 55$ ($q = 0,7$) e $\mathcal{I} \approx 78$ ($q = 1,0$), de acordo com a Figura 22(b). Considerando os cenários para 10 e 20 dimensões, são apresentadas as Figuras 22(c) e 22(d), respectivamente. Nota-se que para 10 dimensões, $\mathcal{I} \approx 145$ ($q = 1,9$) e $\mathcal{I} \approx 180$ ($q = 1,0$), enquanto que com 20 dimensões $\mathcal{I} \approx 190$ ($q = 1,3$) e $\mathcal{I} \approx 230$ ($q = 1,5$). Para o último cenário, tem-se a Figura 22(e) que ilustra que o melhor desempenho em relação ao menor número de iterações necessárias ($\mathcal{I} \approx 220$) é obtido quando $q = 0,1$.



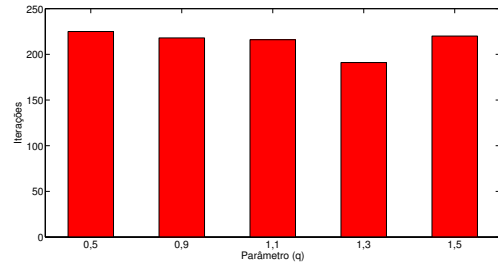
a) Parâmetro q para Função Rastrigin 2-D



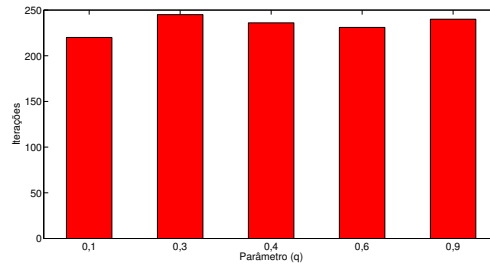
b) Parâmetro q para Função Rastrigin 5-D



c) Parâmetro q Função Rastrigin 10-D



d) Parâmetro q Função Rastrigin 20-D



e) Parâmetro q para Função Rastrigin 50-D

Figura 22: Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Rastrigin considerando diferentes valores para o parâmetro q .

Fonte: Autoria própria

A Figura 23 ilustra o número de iterações necessárias para a convergência do algoritmo β -PSO para a função Rosenbrock com diferentes dimensões e valores para o parâmetro q . Observa-se na Figura 23(a) que a variação do número necessário de iterações para a convergência da função com 2 dimensões variou entre $\mathcal{I} \approx 12$ ($q = 0,9$) e $\mathcal{I} \approx 18$ ($q = 1,9$). Na

Figura 23(b), que ilustra a função Rosenbrock com 5 dimensões, percebe-se que a oscilação variou entre $\mathcal{I} \approx 30$ ($q = 1,3$) e $\mathcal{I} \approx 40$ ($q = 2,1$). As Figuras 23(c) e 23(d) ilustram o número de iterações necessárias para a convergência considerando 10 e 20 dimensões, respectivamente, e nota-se a oscilação entre $\mathcal{I} \approx 63$ ($q = 1,8$) e $\mathcal{I} \approx 80$ ($q = 1,2$), e $\mathcal{I} \approx 170$ ($q = 0,5$), e $\mathcal{I} \approx 200$ ($q = 1,3$). Analisando o cenário com 50 dimensões, observa-se na Figura 23(e) que para o valor do parâmetro $q = 0,2$ o número de iterações foi o menor obtido ($\mathcal{I} \approx 180$).

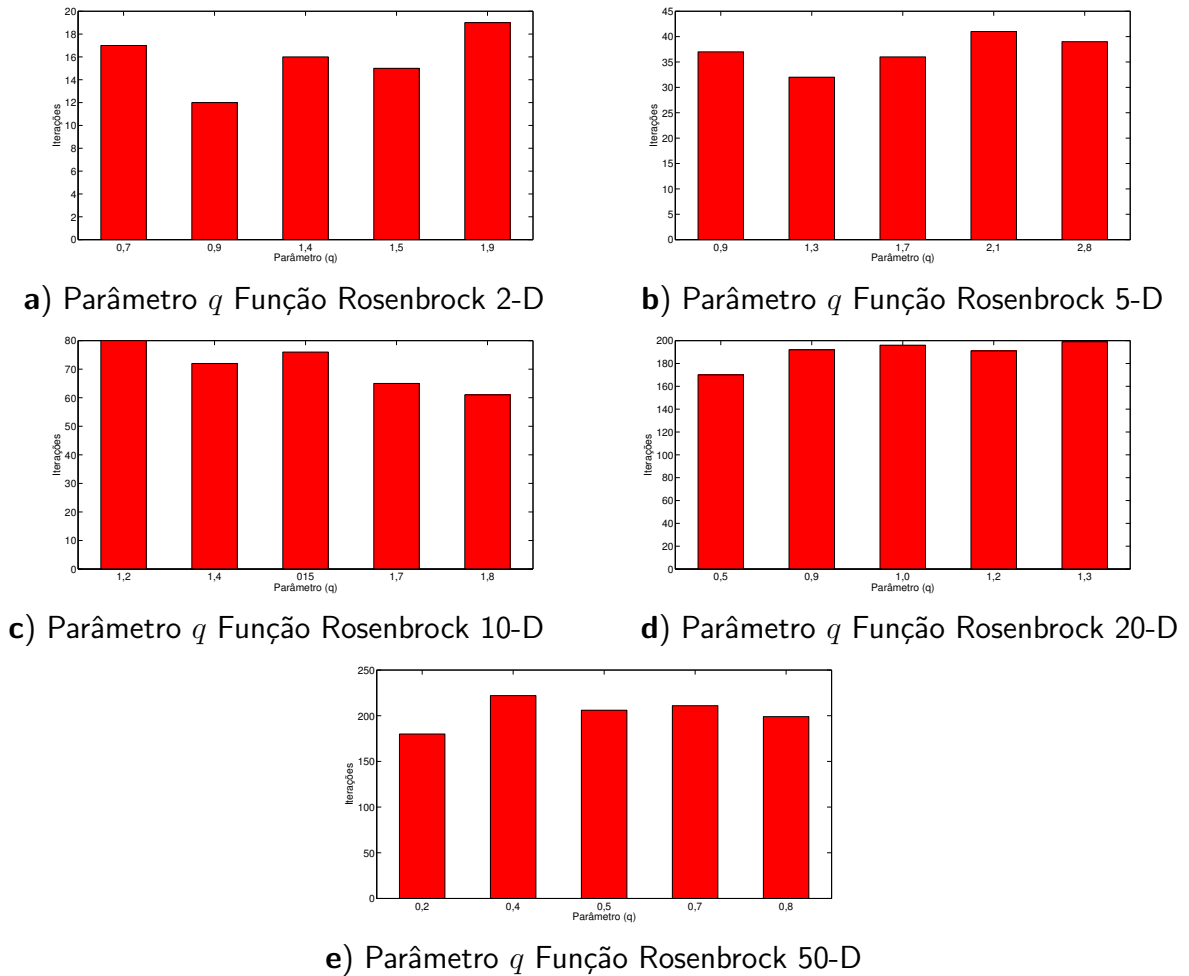


Figura 23: Número de iterações necessárias para a convergência da metaheurística β -PSO com a função Rosenbrock considerando diferentes valores para o parâmetro q .

Fonte: Autoria própria

A Tabela 8 mostra os melhores valores obtidos para o parâmetro q , os quais são utilizados para as simulações das metaheurísticas deste trabalho.

Tabela 8: Parâmetro q da Metaheurística β -PSO para as funções Benchmarks.

Benchmark	Parâmetro (q)				
	Dimensões				
	2	5	10	20	50
Ackley	1,2	1,1	2,2	0,7	0,3
Griewank	1,1	1,0	1,4	1,7	3,1
Rastrigin	1,1	0,7	1,9	1,3	0,1
Rosenbrock	0,9	1,3	1,8	0,5	0,2

3.6.1.3 CALIBRAÇÃO DOS PARÂMETROS DA METAHEURÍSTICA CHAOS-PSO

Para a metaheurística C-PSO, o fator de amortecimento gerado pelo mapa caótico logístico foi calibrado considerando o parâmetro de controle para a obtenção do melhor comportamento caótico e rápida convergência, considerando o menor número necessário de iterações (SANTANA et al., 2018). Dessa maneira, neste trabalho são calibrados os valores para o controle do fator de amortecimento ξ , sendo adotado o intervalo de valores entre $[0, 1; 4]$. A Tabela 9 resume os melhores valores para o parâmetro ξ responsáveis pela melhor convergência e soluções das funções benchmark avaliadas, considerando diferentes dimensões para as mesmas.

Tabela 9: Parâmetro p da Metaheurística C-PSO para as funções Benchmarks.

Benchmark	Parâmetro (ξ)				
	Dimensões				
	2	5	10	20	50
Ackley	2,2	3,4	2,0	2,5	1,1
Griewank	2,0	2,0	1,5	4,0	4,0
Rastrigin	3,0	2,0	1,5	1,0	2,0
Rosenbrock	0,8	1,0	1,5	3,0	4,0

Fonte: Autoria própria

A Figura 24 mostra o número de iterações necessárias para a convergência da metaheurística C-PSO considerando diferentes valores para o fator de amortecimento e dimensões da função Ackley. Pode-se notar na Figura 24(a) que para $\xi = 2,2$ foi obtido o menor número de iterações ($\mathcal{I} \approx 10$). Já os valores $\xi = 3,4$ e $\xi = 2,0$, foram os melhores encontrados para 5 e 10 dimensões, mostrados nas Figuras 24(b) e 24(c), sendo que esses valores apresentaram $\mathcal{I} \approx 23$ e $\mathcal{I} \approx 100$ iterações para a convergência, respectivamente. Já na Figura 24(d), são apresentados os melhores valores do parâmetro ξ para a convergência da função Ackley com 20 dimensões; nota-se que para $\xi = 2,5$ foram necessárias apenas $\mathcal{I} \approx 180$ iterações. Analisando o cenário com 50 dimensões, observa-se que na Figura 24(e) o menor número de iterações

($\mathcal{I} \approx 270$) foi obtido quando $\xi = 1,1$.

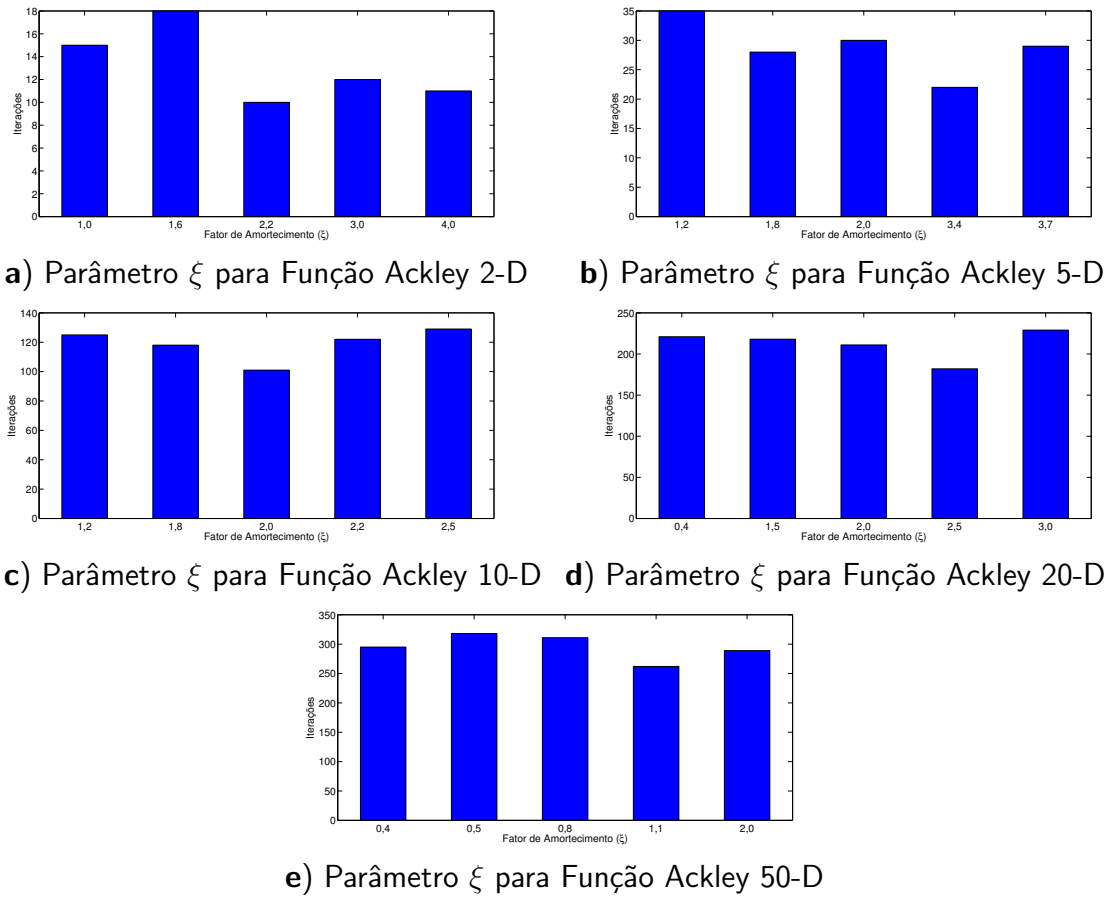


Figura 24: Número de iterações necessárias para a convergência da metaheurística C-PSO com a função Ackley considerando diferentes valores para o fator de amortecimento ξ .

Fonte: Autoria própria

A Figura 25 ilustra o número de iterações necessárias para a convergência do algoritmo C-PSO para a função Griewank com diferentes dimensões e valores para o parâmetro ξ . Observa-se na Figura 25(a) que a variação do número necessário de iterações para a convergência da função com 2 dimensões variou entre $\mathcal{I} \approx 7$ ($\xi = 2,0$) e $\mathcal{I} \approx 17$ ($\xi = 0,9$). Na Figura 25(b), que ilustra a função Griewank com 5 dimensões, percebe-se que a oscilação variou entre $\mathcal{I} \approx 22$ ($\xi = 2,0$) e $\mathcal{I} \approx 40$ ($\xi = 0,5$). As Figuras 25(c) e 25(d) ilustram o número de iterações necessárias para a convergência considerando 10 e 20 dimensões, respectivamente, e nota-se a oscilação entre $\mathcal{I} \approx 75$ ($\xi = 1,5$) e $\mathcal{I} \approx 100$ ($\xi = 3,0$), e $\mathcal{I} \approx 160$ ($q\xi = 4,0$), e $\mathcal{I} \approx 200$ ($\xi = 2,0$). Analisando o cenário com 50 dimensões, observa-se na Figura 25(e) que para o valor do parâmetro $\xi = 4,0$ o número de iterações foi o menor obtido ($\mathcal{I} \approx 190$).

A Figura 26 mostra o número de iterações necessárias para a convergência da me-

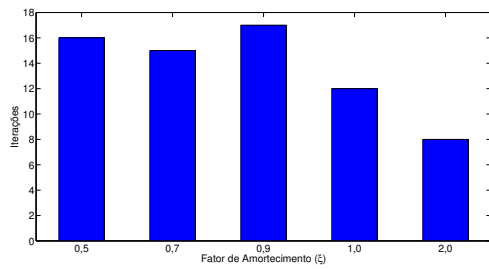
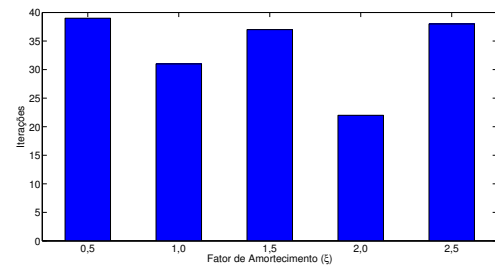
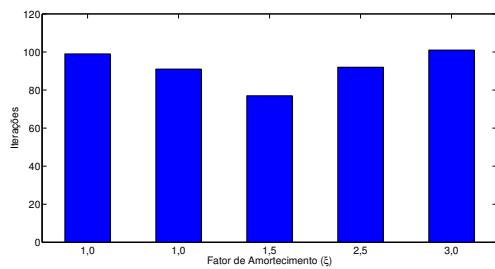
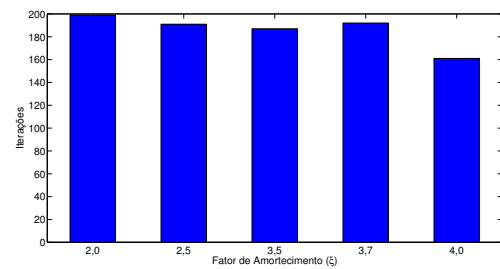
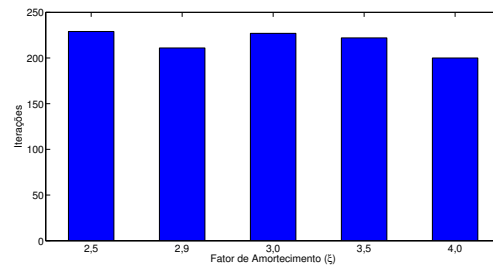
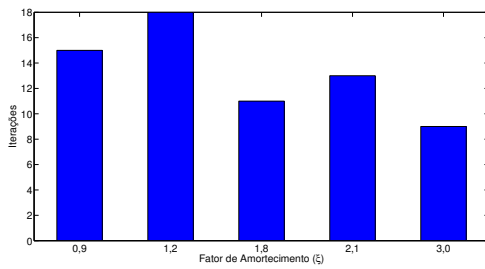
a) Parâmetro ξ para Função Griewank 2-Db) Parâmetro ξ para Função Griewank 5-Dc) Parâmetro ξ Função Griewank 10-Dd) Parâmetro ξ Função Griewank 20-De) Parâmetro ξ para Função Griewank 50-D

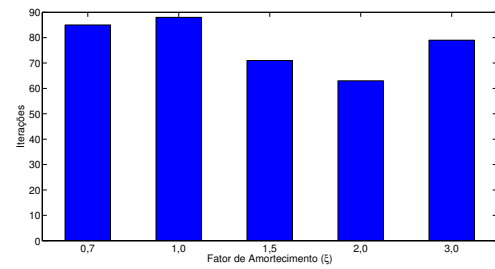
Figura 25: Número de iterações necessárias para a convergência da metaheurística C-PSO com a função Griewank considerando diferentes valores para o fator de amortecimento ξ .

Fonte: Autoria própria

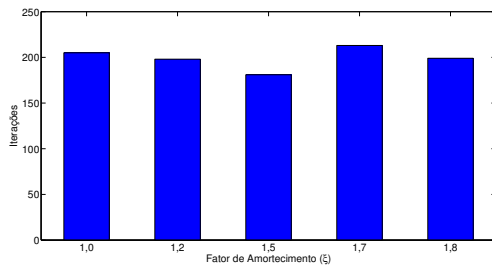
metaheurística C-PSO considerando diferentes valores para o fator de amortecimento e dimensões da função Rastringin. Pode-se notar na Figura 26(a) que para $\xi = 3,0$ foi obtido o menor número de iterações ($\mathcal{I} \approx 9$). Já os valores $\xi = 2,0$ e $\xi = 1,5$, foram os melhores encontrados para 5 e 10 dimensões, mostrados nas Figuras 26(b) e 26(c), sendo que esses valores apresentaram $\mathcal{I} \approx 65$ e $\mathcal{I} \approx 180$ iterações necessárias para a convergência, respectivamente. Já na Figura 26(d) é apresentado o valor do parâmetro ξ para a convergência da função Rastringin com 20 dimensões; nota-se que para $\xi = 1,0$ foram necessárias $\mathcal{I} \approx 180$ iterações. Analisando o cenário com 50 dimensões, observa-se que na Figura 26(e) o menor número de iterações ($\mathcal{I} \approx 230$) foi obtido quando $\xi = 1,2$.



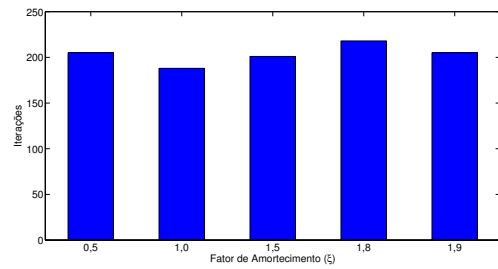
a) Parâmetro ξ para Função Rasringin 2-D



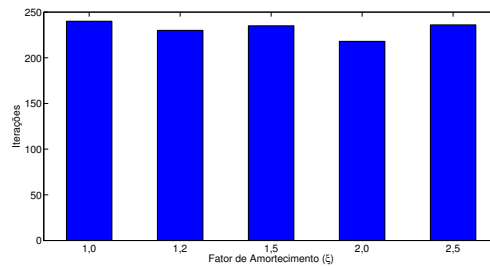
b) Parâmetro ξ para Função Rasringin 5-D



c) Parâmetro ξ Função Rasringin 10-D



d) Parâmetro ξ Função Rasringin 20-D



e) Parâmetro ξ para Função Rasringin 50-D

Figura 26: Número de iterações necessárias para a convergência da metaheurística C-PSO com a função Rastringin considerando diferentes valores para o fator de amortecimento ξ .

Fonte: Autoria própria

A Figura 27 ilustra o número de iterações necessárias para a convergência do algoritmo C-PSO para a função Rosenbrock com diferentes dimensões e valores para o parâmetro ξ . Observa-se na Figura 27(a) que a variação do número necessário de iterações para a con-

vergência da função com 2 dimensões variou entre $\mathcal{I} \approx 8$ ($\xi = 0,8$) e $\mathcal{I} \approx 17$ ($\xi = 1,5$). Na Figura 27(b), que ilustra a função Griewank com 5 dimensões, percebe-se que a oscilação variou entre $\mathcal{I} \approx 27$ ($\xi = 1,0$) e $\mathcal{I} \approx 57$ ($\xi = 2,7$). As Figuras 27(c) e 27(d) ilustram o número de iterações necessárias para a convergência considerando 10 e 20 dimensões, respectivamente, e nota-se a oscilação entre $\mathcal{I} \approx 95$ ($\xi = 1,5$) e $\mathcal{I} \approx 105$ ($\xi = 0,4$), e $\mathcal{I} \approx 140$ ($q\xi = 3,0$), e $\mathcal{I} \approx 190$ ($\xi = 2,9$). Analisando o cenário com 50 dimensões, observa-se na Figura 27(e) que para o valor do parâmetro $\xi = 4,0$ o número de iterações foi o menor obtido ($\mathcal{I} \approx 200$).

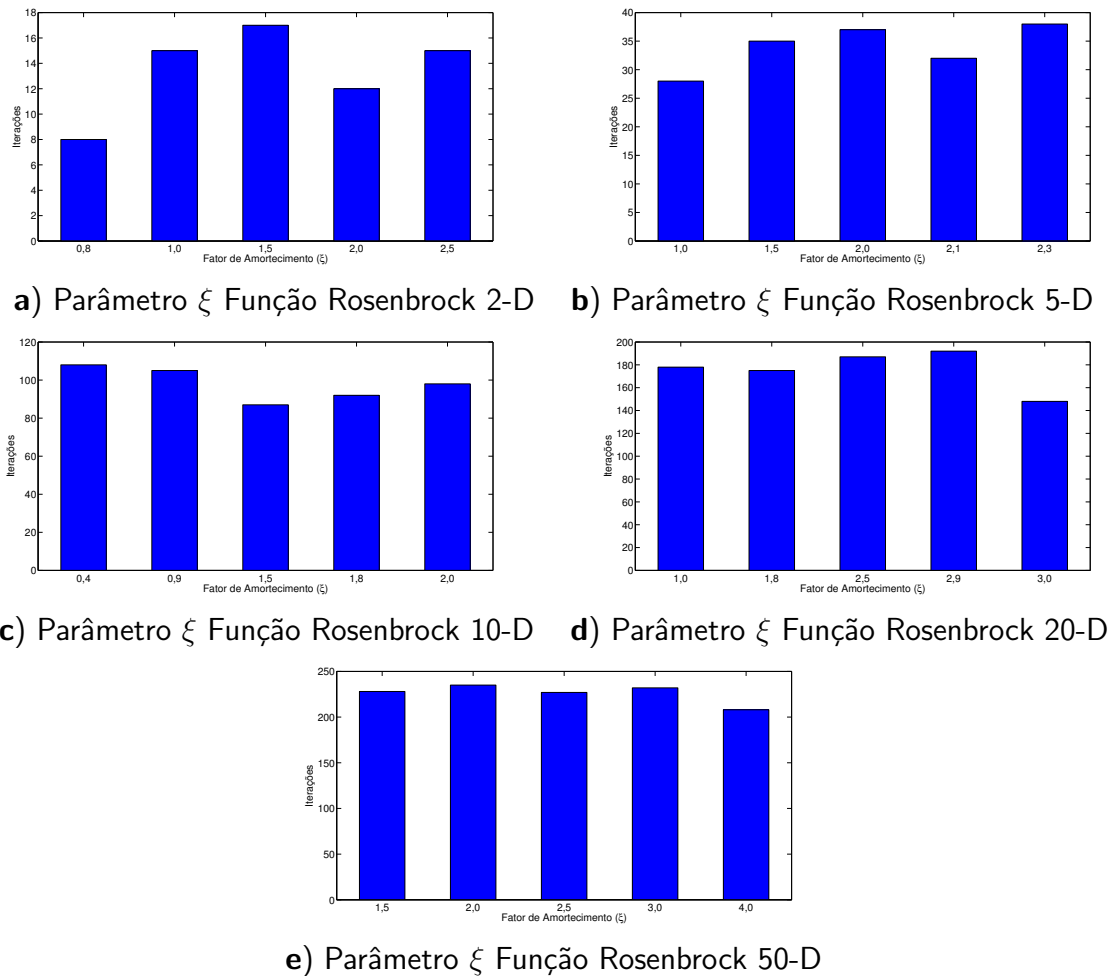


Figura 27: Número de iterações necessárias para a convergência da metaheurística C-PSO com a função Rosenbrock considerando diferentes valores para o fator de amortecimento ξ .

Fonte: Autoria própria

3.6.1.4 CALIBRAÇÃO DOS PARÂMETROS DA METAHEURÍSTICA HALTON-PSO

Neste trabalho, o operador σ_r , referente a combinação dos números primos da expansão b na sequência de Halton tradicional, foi gerado através da utilização do algoritmo

B&W (DIGEHSARA et al., 2020), e a permutação empregada foi adaptada de Digehsara et al. (2020). A seguir são apresentados os resultados da calibração, considerando o número de iterações necessárias para a convergência da metaheurística H-PSO e diferentes valores para o parâmetro b , porém apenas os valores mais representativos são ilustrados nesta Seção. O intervalo de valores adotado variou entre $[2, 29]$. A Tabela 10 resume os melhores valores do parâmetro b encontrados para cada função Benchmark considerando diferentes dimensões.

Tabela 10: Parâmetro p da Metaheurística H-PSO para as funções Benchmarks.

Benchmark	Parâmetro (b)				
	Dimensões				
	2	5	10	20	50
Ackley	11	3	17	11	7
Griewank	13	3	13	7	23
Rastrigin	13	3	11	11	23
Rosenbrock	17	3	17	23	19

Na Figura 28, é observada a variação do número de iterações necessárias para a convergência da metaheurística H-PSO. Na Figura 28(a) tem-se a função Ackley com 2 dimensões e percebe-se que a oscilação se deu entre $\mathcal{I} \approx 10$ ($b = 11$) e $\mathcal{I} \approx 18$ ($b = 17$). Esse comportamento está relacionado à variação das permutações para a alteração do número primo como base. Para esse caso, o valor $b = 11$ representou o melhor compromisso entre diversificação e exploração para a inicialização da população do algoritmo. Já as Figuras 28(b) e 28(c) apresentam como melhores valores para base $b = 3$ e $b = 17$, os quais obtiveram $\mathcal{I} \approx 20$ e $\mathcal{I} \approx 120$ iterações, para as dimensões 5 e 10, respectivamente. Para o cenário com 20 e 50 dimensões, o melhor compromisso foi apresentado por $\mathcal{I} \approx 180$ ($b = 11$) e $\mathcal{I} \approx 250$ ($b = 7$), conforme ilustrado nas Figuras 28(d) e 28(e), respectivamente.

Na Figura 29, é ilustrado o comportamento do algoritmo H-PSO para a função Griewank com diferentes dimensões e valores para o parâmetro b . Nota-se que a variação desse parâmetro pode influenciar a velocidade de convergência, já que a utilização da sequência de Halton Embaralhada consegue melhorar o processo de inicialização das partículas do enxame no espaço de busca. De acordo com a Figura 29(a), a variação do número necessário de iterações para a convergência da função Griewank com 2 dimensões variou entre $\mathcal{I} \approx 10$ ($b = 13$) e $\mathcal{I} \approx 16$ ($b = 7$). O mesmo comportamento é observado nas Figuras 29(b) e 29(c), nas quais a faixa de variação das iterações foi de $\mathcal{I} \approx 32$ ($b = 3$) e $\mathcal{I} \approx 45$ ($b = 7$), para 5 dimensões, e $\mathcal{I} \approx 75$ ($b = 13$) e $\mathcal{I} \approx 105$ ($b = 17$), para 10 dimensões. Já para 20 e 50 dimensões, ilustradas nas Figuras 29(d) e 29(e), respectivamente, percebe-se que os melhores desempenhos foram obtidos quando $b = 7$ e $b = 23$.

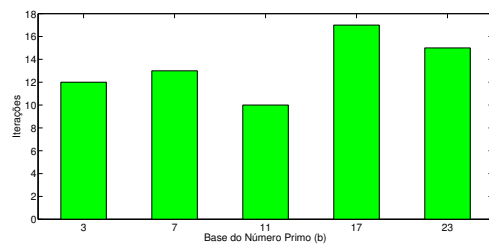
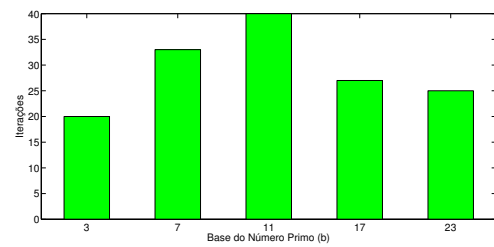
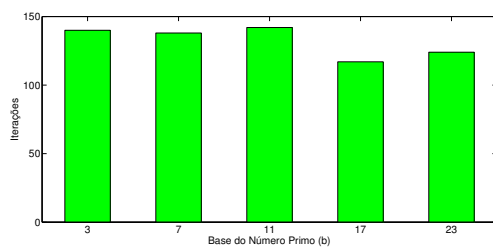
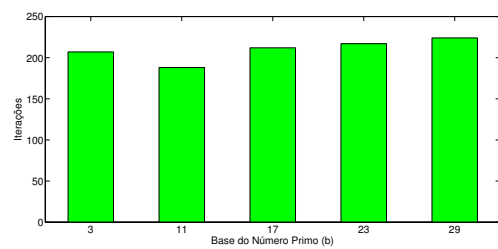
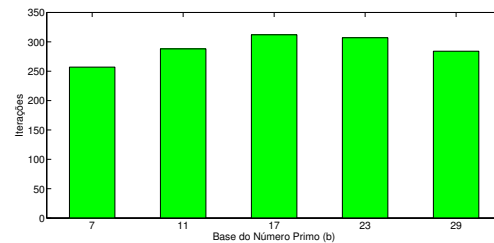
a) Parâmetro b para Função Ackley 2-Db) Parâmetro b para Função Ackley 5-Dc) Parâmetro b para Função Ackley 10-Dd) Parâmetro b para Função Ackley 20-De) Parâmetro b para Função Ackley 50-D

Figura 28: Número de iterações necessárias para a convergência da metaheurística H-PSO com a função Ackley considerando diferentes valores para a base do número primo b .

Fonte: Autoria própria

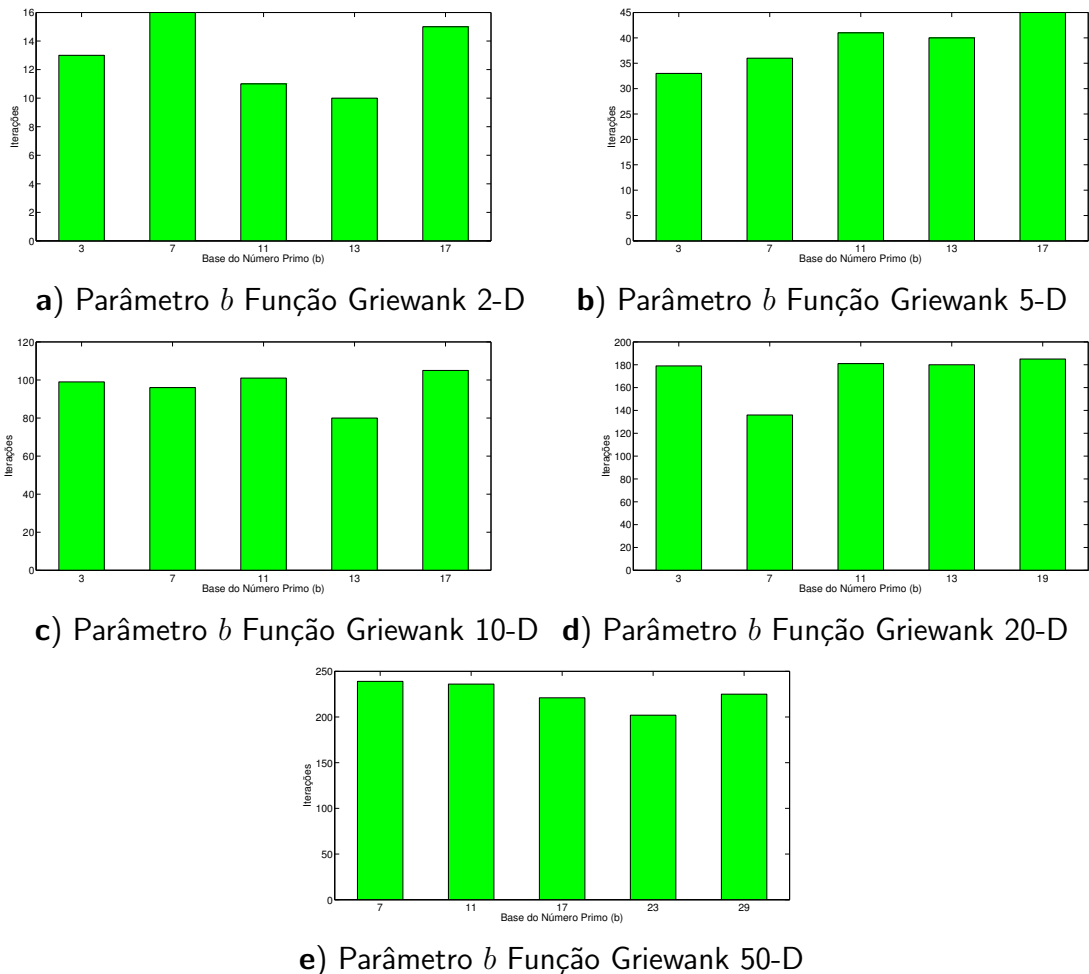


Figura 29: Número de iterações necessárias para a convergência da metaheurística H-PSO com a função Griewank considerando diferentes valores para a base do número primo b .

Fonte: Autoria própria

Para a função Rastrigin é realizada calibração do parâmetro b considerando diferentes dimensões, conforme ilustrado na Figura 30. Assim como nas funções anteriores, a variação desse parâmetro também influenciou a velocidade de convergência da metaheurística H-PSO. Nas Figuras 30(a) e 30(b) a variação oscilou entre $\mathcal{I} \approx 5$ ($b = 13$) e $\mathcal{I} \approx 16$ ($b = 3$) e $\mathcal{I} \approx 65$ ($b = 3$) e $\mathcal{I} \approx 87$ ($b = 7$), considerando 2 e 5 dimensões, respectivamente. As Figuras 30(c), 30(d) e 30(e) mostram a variação do parâmetro b para a função Rastrigin com 10, 20 e 50 dimensões, nas quais observa-se que os melhores valores para esse parâmetro corresponderam a $b = 11$, $b = 11$ e $b = 23$, respectivamente.

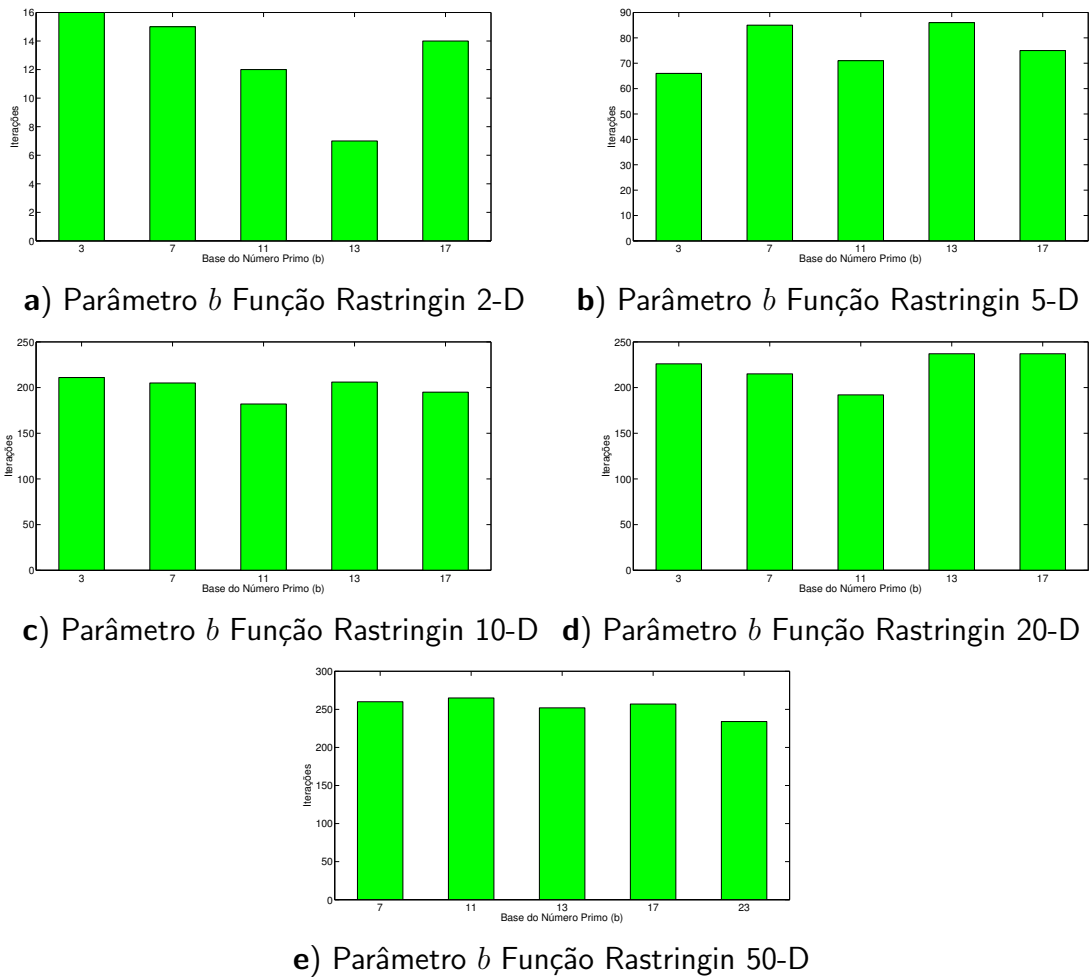
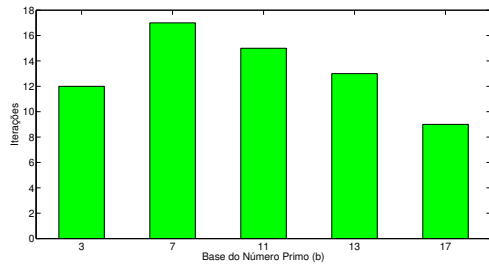


Figura 30: Número de iterações necessárias para a convergência da metaheurística H-PSO com a função Rastrigin considerando diferentes valores para a base do número primo b .

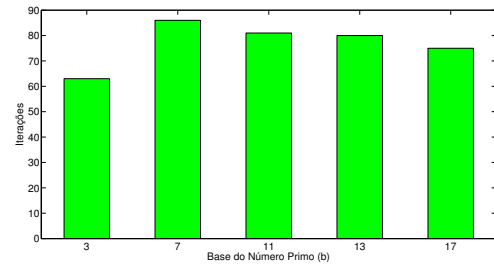
Fonte: Autoria própria

A Figura 31 ilustra a variação do número de iterações necessárias para a convergência da metaheurística H-PSO, considerando diferentes valores para o parâmetro b e diferentes dimensões para a função Rosenbrock. Na Figura 31(a), tem-se a função com 2 dimensões e nota-se a oscilação do número de iterações em relação ao parâmetro b entre $\mathcal{I} \approx 8$ ($b = 17$) e

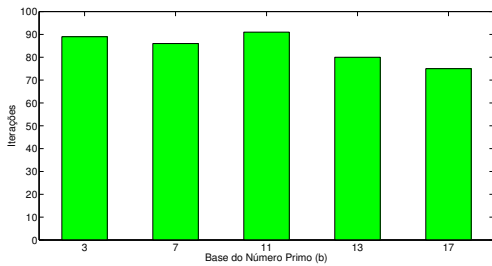
$\mathcal{I} \approx 18$ ($b = 7$). Já as Figuras 31(b) e 31(c) apresentam como melhores valores para o referido parâmetro $b = 3$ e $b = 17$, os quais obtiveram $\mathcal{I} \approx 60$ e $\mathcal{I} \approx 75$ iterações para convergência, considerando as dimensões 5 e 10, respectivamente. Para o cenário com 20 e 50 dimensões, o melhor compromisso foi apresentado por $\mathcal{I} \approx 135$ ($b = 23$) e $\mathcal{I} \approx 170$ ($b = 19$), conforme ilustrado nas Figuras 31(d) e 31(e), respectivamente.



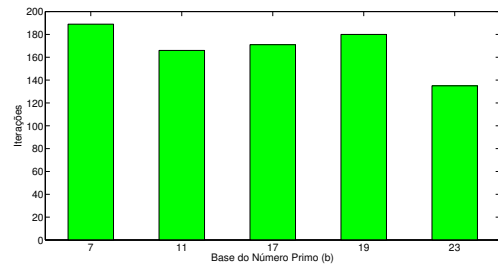
a) Parâmetro b Função Rosenbrock 2-D



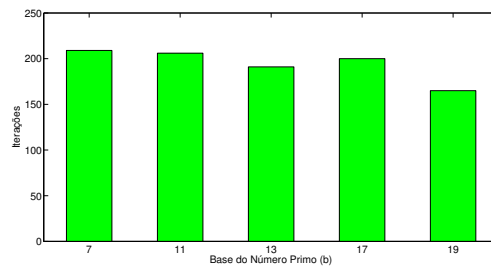
b) Parâmetro b Função Rosenbrock 5-D



c) Parâmetro b Função Rosenbrock 10-D



d) Parâmetro b Função Rosenbrock 20-D



e) Parâmetro b Função Rosenbrock 50-D

Figura 31: Número de iterações necessárias para a convergência da metaheurística H-PSO com a função Rosenbrock considerando diferentes valores para a base do número primo b .

Fonte: Autoria própria

3.7 COMPARAÇÃO ENTRE AS METAHEURÍSTICAS

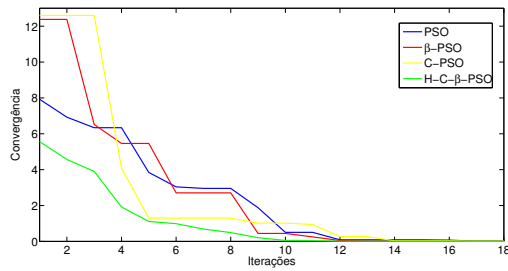
Nesta seção, as metaheurísticas apresentadas neste trabalho são avaliadas e validadas. A abordagem proposta Halton-Chaos- β -PSO é comparada em termos de qualidade e velocidade de convergência, número de iterações e complexidade computacional, com três metaheurísticas: a) PSO tradicional, b) β -PSO e c) C-PSO.

3.7.1 SIMULAÇÕES E RESULTADOS

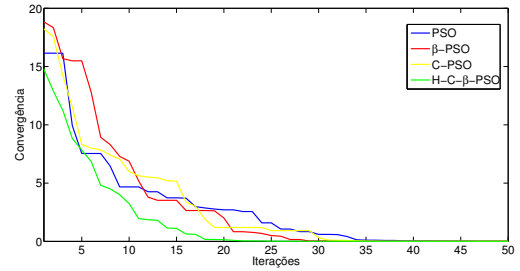
Para a execução das metaheurísticas foi utilizado o software Matlab, considerando os parâmetros previamente descritos na Seção 3.6. Para todos os cenários e funções Benchmark foi considerado o critério de parada definido em 2.5.1. Assim, o valor mínimo obtido pelas metaheurísticas, quando as mesmas convergiram, foi de $10e^{-2}$. A seguir, os resultados numéricos da média de 50 realizações de cada metaheurística são apresentados e discutidos.

A convergência das metaheurísticas para a função Ackley é apresentada na Figura 32, na qual pode-se observar que a metaheurística proposta obteve convergência mais rápida para todos os cenários avaliados, exigindo apenas $\mathcal{I} \approx 10$ para $d = 2$, enquanto o PSO tradicional necessitou de $\mathcal{I} \approx 18$, conforme 32(a). Para $d = 5$ e $d = 10$ são apresentadas as Figuras 32(b) e 32(c), nas quais pode-se observar que a velocidade de convergência da metaheurística proposta foi a menor dentre os algoritmos, necessitando de $\mathcal{I} \approx 20$ e $\mathcal{I} \approx 35$, respectivamente. Para $d = 20$, o comportamento do algoritmo Halton-Chaos- β -PSO também se mostrou superior, necessitando de $\mathcal{I} \approx 100$ para a convergência; já a metaheurística β -PSO, apesar de apresentar grande oscilação, conseguiu alcançar a convergência com $\mathcal{I} \approx 150$, sendo mais rápida que o PSO tradicional ($\mathcal{I} \approx 240$). A metaheurística C-PSO, apesar de ser um pouco mais lenta que a abordagem proposta, demonstrou um bom desempenho de convergência e menor oscilação que os algoritmos PSO tradicional e β -PSO para cenários com maiores dimensões, como por exemplo, para $d = 20$ e $d = 50$. A Figura 32(e) mostra que todas as metaheurísticas utilizadas, quando bem calibradas, são adequadas para maiores dimensões ($d = 50$), sendo que o menor número de iterações necessárias para a convergência foi obtido pelo H-C- β -PSO ($\mathcal{I} \approx 120$), seguido pelas metaheurísticas β -PSO ($\mathcal{I} \approx 140$), C-PSO ($\mathcal{I} \approx 160$) e PSO tradicional ($\mathcal{I} \approx 270$).

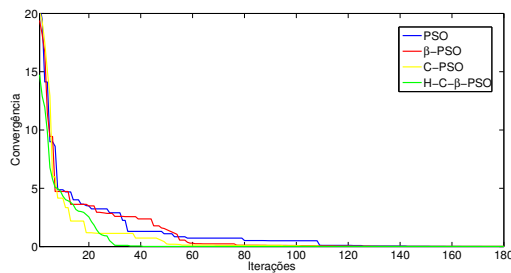
Além do menor número de iterações necessárias para convergência com a função Ackley, a metaheurística proposta H-C- β -PSO também apresentou os melhores resultados em relação à média, mediana e desvio padrão, ilustrados na Tabela 11. Percebe-se que para todos os cenários, a metaheurística proposta alcançou o menor valor de desvio padrão, inclusive para $d = 50$. Adicionalmente, os valores da mediana indicaram que, para todos os cenários avaliados, o algoritmo proposto praticamente alcançou o valor estipulado para a convergência com um número bem menor de iterações que o PSO tradicional. Além disso, como a população inicial é definida e calibrada utilizando a sequência de Halton, os valores de máximo da metaheurística proposta se mostraram inferiores dos demais algoritmos, fator este que também auxiliou a convergência mais rápida e a qualidade dos resultados obtidos. O algoritmo β -PSO apresentou valores de máximos, médias e medianas maiores que os outros algoritmos (inclusive



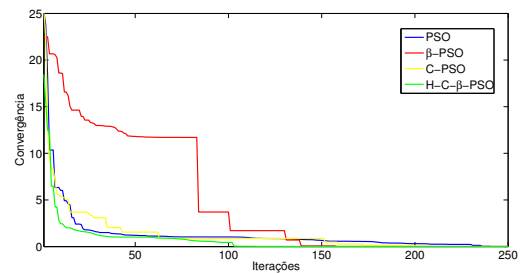
a) Convergência - Função Ackley 2-D



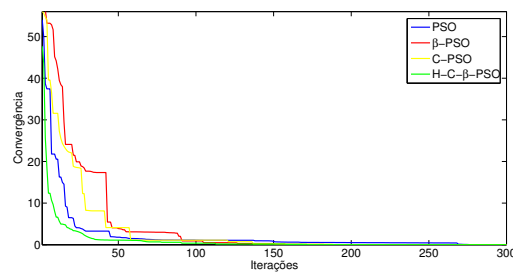
b) Convergência - Função Ackley 5-D



c) Convergência - Função Ackley 10-D



d) Convergência - Função Ackley 20-D



e) Convergência - Função Ackley 50-D

Figura 32: Número de iterações necessárias para a convergência das metaheurísticas com a função Ackley.

Fonte: Autoria própria

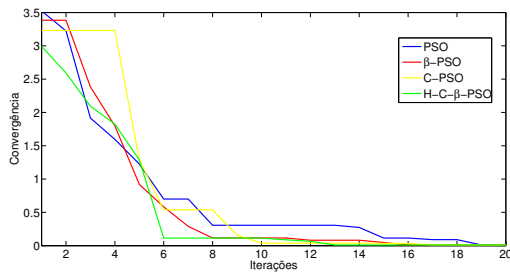
o PSO tradicional) devido à sua maior capacidade de exploração (diversificação), já que proporciona maior abrangência do espaço de busca. Entretanto, sua convergência ainda permaneceu mais rápida que o PSO tradicional, além da maior qualidade dos resultados obtidos (demonstrado nas próximas seções). O mesmo comportamento foi observado para a metaheurística C-PSO, que mesmo obtendo convergência mais rápida que o PSO tradicional, apresentou alguns resultados inferiores; porém isso pode ser justificado devido à sua maior capacidade de exploração, na tentativa de evitar a convergência prematura para mínimos locais.

Tabela 11: Resultados numéricos das Metaheurísticas para a função Ackley.

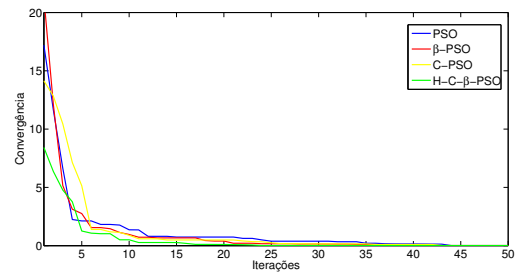
Heurística	Dimensão	Resultados			
		Média	Mediana	Máximo	StD
PSO	2	2,1843	0,5054	7,9272	2,7275
	5	3,0070	1,3269	16,1341	4,1576
	10	1,4033	0,5017	20,8311	2,9012
	20	1,3435	0,8592	25,1664	2,7666
	50	2,4565	0,7405	54,0345	6,6794
β-PSO	2	2,5803	0,3475	12,3787	3,9596
	5	3,2975	0,4744	18,8310	5,3432
	10	1,2894	0,0986	19,3844	2,8326
	20	5,0251	1,7047	22,4984	6,4134
	50	4,7581	0,2009	56,4587	11,3838
C-PSO	2	2,5372	0,9800	12,6024	4,4382
	5	3,0819	0,9212	18,2351	4,5620
	10	0,9245	0,1039	20,3389	2,9338
	20	1,3454	0,8741	25,3529	2,6267
	50	3,5793	0,2295	56,0903	9,2225
HC-β-PSO	2	0,9812	0,0445	5,5662	1,6935
	5	1,7817	0,0129	14,7698	3,5643
	10	0,7027	0,0100	14,7354	2,1545
	20	0,7248	0,0201	18,4597	2,0104
	50	1,1555	0,1555	47,6947	4,3967

A Figura 33 ilustra o número de iterações necessárias para a convergência das metaheurísticas validadas para a função Griewank. Nas Figuras 33(a) e 33(b), observa-se que a metaheurística proposta obteve convergência mais rápida, exigindo $\mathcal{I} \approx 12$ e $\mathcal{I} \approx 20$, para $d = 2$ e $d = 5$, respectivamente, enquanto o algoritmo PSO tradicional necessitou de $\mathcal{I} \approx 19$ e $\mathcal{I} \approx 43$. Para esses cenários com poucas dimensões, as metaheurísticas β -PSO e C-PSO também obtiveram bom desempenho, necessitando de um número menor de iterações para convergência quando comparadas ao PSO tradicional. A convergência das metaheurísticas para a função Griewank também foi avaliada para $d = 10$, $d = 20$ e $d = 50$, conforme Figuras 33(c), 33(d) e 33(e), respectivamente. Para esses cenários, a abordagem proposta necessitou

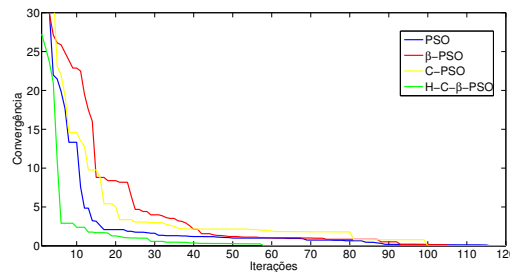
de um número menor de iterações para a convergência, especialmente para $d = 50$, onde precisou de $\mathcal{I} \approx 95$, quase 30% a menos da quantidade de iterações necessárias para a convergência do algoritmo tradicional ($\mathcal{I} \approx 285$). Foi observado que para esta função Benchmark o desempenho dos algoritmos β -PSO e C-PSO foram satisfatórios, apresentando poucas oscilações e menor número de iterações necessárias para a convergência que o PSO tradicional. Na Tabela 12, pode-se observar que a metaheurística H-C- β -PSO obteve os melhores resultados para todas as dimensões da função Griewank, especialmente para dimensões mais altas, onde os valores do desvio padrão e média foram consideravelmente menores. Para $d = 50$, os resultados dos algoritmos C-PSO e β -PSO demonstraram notável superioridade em relação ao PSO tradicional, se mostrando mais adequados para problemas com maiores dimensões, assim como a metaheurística proposta.



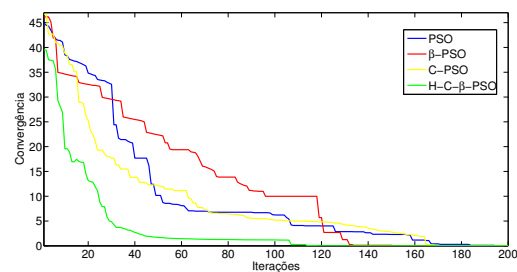
a) Convergência - Função Griewank 2-D



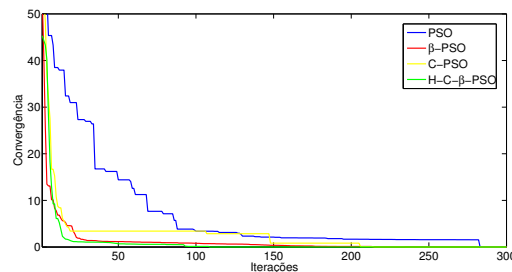
b) Convergência - Função Griewank 5-D



c) Convergência - Função Griewank 10-D



d) Convergência - Função Griewank 20-D



e) Convergência - Função Griewank 50-D

Figura 33: Número de iterações necessárias para a convergência das metaheurísticas com a função Griewank.

Fonte: Autoria própria

Na Figura 34, é observado que a metaheurística H-C- β -PSO se mostra como a mais

Tabela 12: Resultados numéricos das Metaheurísticas para a função Griewank.

Heurística	Dimensão	Resultados			
		Média	Mediana	Máximo	STD
PSO	2	0,2902	0,2800	2,1024	1,6120
	5	1,2919	0,3829	17,2022	2,9543
	10	2,6989	0,9889	30,7429	6,0298
	20	10,3293	6,2235	44,9533	12,6665
	50	7,5212	2,1069	53,8475	11,2897
β-PSO	2	0,2172	0,2178	2,3727	1,1120
	5	1,1931	0,1663	21,2749	3,4556
	10	4,4894	1,0415	31,9076	7,7730
	20	12,7005	9,9879	46,1642	12,8023
	50	1,2953	0,3668	53,8606	4,5139
C-PSO	2	0,2522	0,3700	3,1024	1,4101
	5	1,3023	0,2140	14,1459	3,1220
	10	3,9908	1,8701	35,1268	6,9817
	20	9,2632	5,1491	46,6919	10,8337
	50	2,7688	0,8291	54,1750	6,2442
HC-β-PSO	2	0,1191	0,1512	4,1007	0,9172
	5	0,6237	0,0101	8,4222	1,6581
	10	1,3312	0,0183	27,2715	4,4799
	20	3,5938	1,1864	39,4847	8,0071
	50	1,1443	0,0100	45,3251	5,4186

adequada para a busca do ponto mínimo da função Rastrigin, apresentando o menor número de iterações para todas as dimensões avaliadas. As Figuras 34(a) e 34(b) mostram que para a convergência do algoritmo proposto foram necessárias $\mathcal{I} \approx 9$ e $\mathcal{I} \approx 22$, para $d = 2$ e $d = 5$, respectivamente. Para as mesmas dimensões, o algoritmo PSO tradicional requereu $\mathcal{I} \approx 13$ e $\mathcal{I} \approx 82$, apresentando a maior quantidade de iterações conforme o aumento das dimensões. As metaheurísticas β -PSO e C-PSO necessitaram de $\mathcal{I} \approx 11$ e $\mathcal{I} \approx 32$ e $\mathcal{I} \approx 11$ e $\mathcal{I} \approx 24$, para $d = 2$ e $d = 5$, respectivamente. As Figuras 34(c), 34(d) e 34(e) mostram o número de iterações necessárias para a convergência das metaheurísticas para a função Rastrigin para $d = 10$, $d = 20$ e $d = 50$, respectivamente. Para esses cenários, a abordagem proposta necessitou de um número menor de iterações para a convergência, especialmente para $d = 50$, no qual precisou de $\mathcal{I} \approx 95$ para a convergência, enquanto o algoritmo PSO tradicional necessitou de ($\mathcal{I} \approx 278$). Para esta função Benchmark, os algoritmos β -PSO e C-PSO obtiveram bom desempenho de convergência, além de razoáveis médias, medianas e desvios padrões. Como citado anteriormente, a metaheurística proposta utiliza a sequência de Halton Embaralhada para a inicialização das partículas do enxame, na qual os parâmetros foram calibradas de acordo com cada função Benchmark e suas respectivas dimensões e, desta maneira, o processo de busca já foi iniciado com uma população mais adequada e com maior probabilidade da obtenção de melhores resultados, conforme ilustrado na Tabela 13.

A Figura 35 ilustra o número de iterações necessárias para a convergência das metaheurísticas validadas para a função Rosenbrock com diferentes dimensões. Em 35(a) e 35(b) pode-se observar que a metaheurística proposta obteve convergência mais rápida, exigindo apenas $\mathcal{I} \approx 10$ para $d = 2$, enquanto o PSO tradicional necessitou de $\mathcal{I} \approx 17$. Para $d = 10$ e $d = 20$ são apresentadas as Figuras 35(b) e 35(c), nas quais pode-se observar que a velocidade de convergência da metaheurística proposta foi a menor dentre os algoritmos, necessitando de $\mathcal{I} \approx 28$ e $\mathcal{I} \approx 25$, respectivamente. Para $d = 20$ o comportamento do algoritmo Halton-Chaos- β -PSO também se mostrou superior, necessitando de $\mathcal{I} \approx 60$ para a convergência; já a metaheurística β -PSO, apesar de apresentar grande oscilação, conseguiu alcançar a convergência com $\mathcal{I} \approx 100$, sendo mais rápida que o PSO tradicional ($\mathcal{I} \approx 185$). A metaheurística C-PSO, apesar de ser um pouco mais lenta que a abordagem proposta, demonstrou um bom desempenho de convergência e menor oscilação que os algoritmos PSO tradicional e β -PSO para cenários com maiores dimensões, como por exemplo, para $d = 20$ e $d = 50$. A Figura 35(e) mostra que todas as metaheurísticas utilizadas, quando bem calibradas, são adequadas para maiores dimensões ($d = 50$), sendo que o menor número de iterações necessárias para a convergência foi obtido pelo H-C- β -PSO ($\mathcal{I} \approx 110$), seguido pelas metaheurísticas β -PSO ($\mathcal{I} \approx 140$), C-PSO ($\mathcal{I} \approx 180$) e PSO tradicional ($\mathcal{I} \approx 270$).

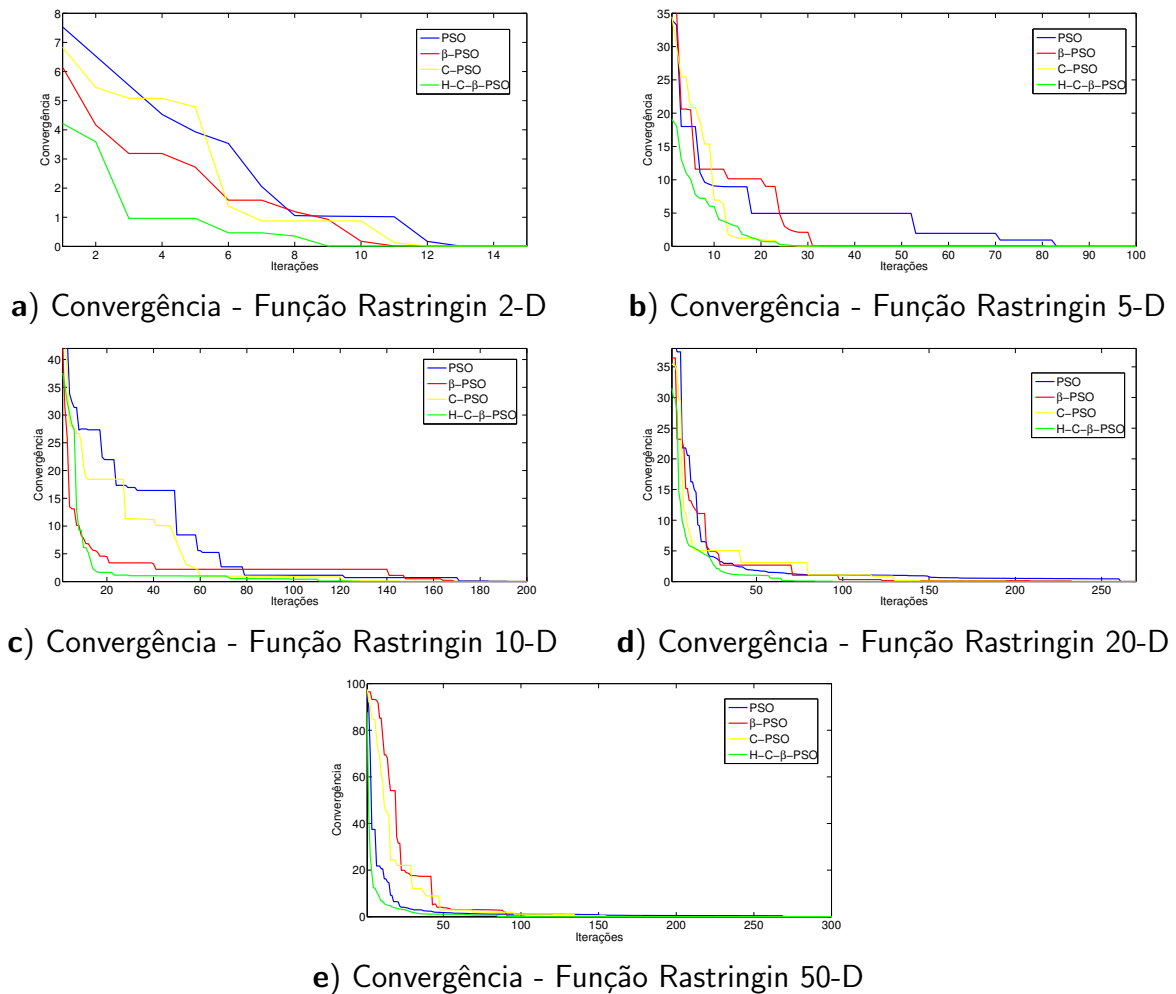


Figura 34: Número de iterações necessárias para a convergência das metaheurísticas com a função Rastrigin.

Fonte: Autoria própria

Tabela 13: Resultados numéricos das Metaheurísticas para a função Rastrigin.

heurística	Dimensão	Resultados			
		Média	Mediana	Máximo	STD
PSO	2	2,5326	1,0586	7,5310	2,5461
	5	4,6134	4,9548	34,0137	5,7870
	10	6,7077	1,1386	43,8475	9,9361
	20	2,5800	1,0047	39,0345	6,3730
	50	2,7921	0,7405	95,0345	9,6214
β-PSO	2	1,6635	1,3811	6,1514	1,8672
	5	3,4637	0,1024	35,6302	6,8580
	10	2,6830	2,2126	42,9184	4,4565
	20	2,1336	0,1546	36,4587	5,3696
	50	7,0278	0,0136	96,4587	12,7279
C-PSO	2	2,1526	0,8743	6,8311	2,4855
	5	2,4104	0,0104	34,5019	6,8506
	10	4,7777	0,8945	44,1750	8,4180
	20	2,1713	0,1871	35,5903	5,2004
	50	5,7535	0,5073	96,390	12,2018
H-C-β-PSO	2	0,8035	0,3545	4,2153	1,3178
	5	1,3376	0,0100	19,0737	3,5065
	10	1,6625	0,4235	37,5812	5,5632
	20	1,1030	0,0101	39,4840	3,9813
	50	1,2365	0,0106	87,6947	6,1022

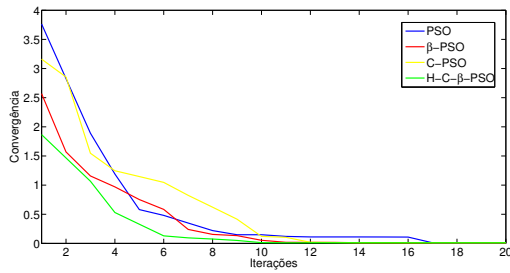
A metaheurística proposta H-C- β -PSO também apresentou os melhores resultados em relação à média, mediana e desvio padrão para a função Rosenbrock, ilustrados na Tabela 14. Para todas as dimensões, a metaheurística proposta alcançou o menor valor de desvio padrão, média e mediana. Como explicado anteriormente, a população inicial desse algoritmo é definida e calibrada utilizando a sequência de Halton e, dessa maneira, os valores de máximo são menores que os dos demais algoritmos, fator este que também auxilia a convergência mais rápida, assim como a qualidade dos resultados obtidos. Os algoritmos β -PSO e C-PSO apresentaram, na maioria dos casos, resultados inferiores que o algoritmo PSO tradicional, sendo este fato justificado por sua maior capacidade de exploração e abrangência do espaço de busca. Entretanto, o número de iterações necessárias para a convergência desses algoritmos foi consideravelmente menor que o PSO tradicional.

Um importante fato que foi observado em todos os cenários para todas as funções Benchmark, é que para a dimensão $d = 20$, o comportamento de todas as metaheurísticas obtiveram pior desempenho que quando avaliadas com $d = 50$. Assim, para os cenários observados neste trabalho, pode-se concluir que para dimensões médias os algoritmos metaheurísticos obtiveram pior desempenho que quando avaliados com pequenas ou grandes dimensões.

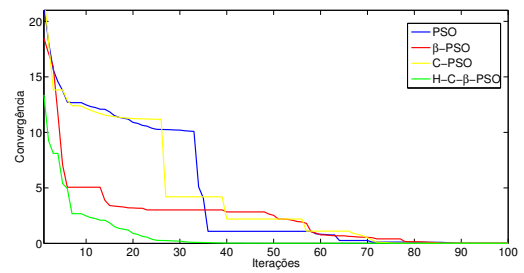
3.7.2 ANÁLISE DA COMPLEXIDADE COMPUTACIONAL

A avaliação da complexidade computacional das metaheurísticas foi realizada em termos de pontos flutuantes (FLOPs), e a qualidade de convergência foi medida através da robustez algorítmica e do erro quadrático médio normalizado. A quantidade exigida de FLOPs das metaheurísticas para as funções Benchmark, considerando diferentes dimensões, está ilustrada nas Tabelas 15, 16, 17 e 18.

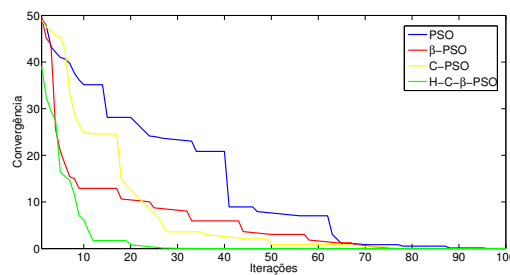
As Figuras 36, 37, 38 e 39 mostram a evolução do tempo de processamento em termos do número de FLOPs consumidos para a convergência das metaheurísticas PSO, β -PSO, C-PSO e H-C- β -PSO para as funções Ackley, Griewank, Rastrigin e Rosenbrock, respectivamente. Em geral, para pequenas dimensões, o número de FLOPs consumidos pelas metaheurísticas foi muito similar. Por outro lado, com o aumento das dimensões das funções Benchmark, percebeu-se que para a metaheurística proposta o número de FLOPs caiu em aproximadamente 30% em relação ao PSO tradicional, devido ao menor número de iterações necessárias para a convergência do algoritmo H-C- β -PSO. Na maioria dos cenários, a metaheurística PSO tradicional se demonstrou como o algoritmo mais complexo, exigindo o maior número de FLOPs para as dimensões consideradas. Em alguns casos, principalmente



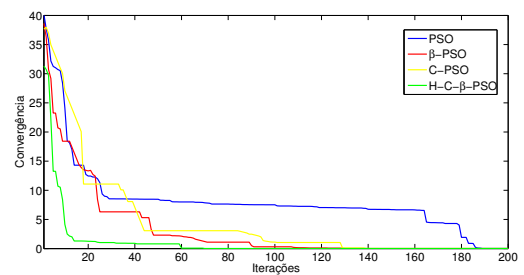
a) Convergência - Função Rosenbrock 2-D



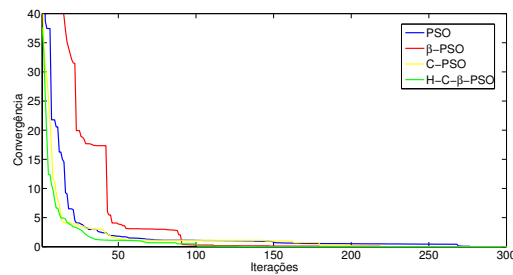
b) Convergência - Função Rosenbrock 5-D



c) Convergência - Função Rosenbrock 10-D



d) Convergência - Função Rosenbrock 20-D



e) Convergência - Função Rosenbrock 50-D

Figura 35: Número de iterações necessárias para a convergência das metaheurísticas com a função Rosenbrock.

Fonte: Autoria própria

Tabela 14: Resultados numéricos das Metaheurísticas para a função Rosenbrock.

Heurística	Dimensão	Resultados			
		Média	Mediana	Máximo	STD
PSO	2	0,6156	0,1345	3,7643	1,0370
	5	4,3985	1,0862	21,5080	5,6220
	10	13,7547	7,5813	49,2264	14,2115
	20	9,1654	12,5340	57,1091	12,4941
	50	2,4035	0,7405	49,0345	6,5299
β-PSO	2	0,4155	0,0361	2,5696	0,6845
	5	2,5030	2,3597	18,4958	3,2264
	10	3,4617	0,3046	36,3045	6,7396
	20	6,1736	3,0351	49,6793	8,9072
	50	5,0185	0,1106	56,4587	12,0777
C-PSO	2	0,6599	0,1122	3,1620	0,9476
	5	4,3335	2,1940	22,4591	5,2304
	10	4,5004	1,0379	37,9691	6,9733
	20	7,4484	0,8683	48,9775	12,7441
	50	1,7122	0,9730	46,5903	4,7288
H-C-β-PSO	2	0,2861	0,0115	1,8663	0,5408
	5	0,8175	0,0111	13,3823	2,1146
	10	0,9924	0,0029	31,7335	3,6660
	20	2,2230	0,0176	39,0877	6,8916
	50	1,0599	0,0104	37,6947	3,7554

Tabela 15: Número de FLOPs para a função Ackley considerando o aumento das dimensões.

Metaheurísticas	FLOPs				
	$d = 2$	$d = 5$	$d = 10$	$d = 20$	$d = 50$
PSO	1,784e3	7,812e3	9,527e4	3,801e5	1,812e6
β-PSO	1,091e3	6,187e3	6,223e4	2,395e5	1,032e6
C-PSO	1,412e3	6,801e3	8,149e4	3,233e5	1,399e6
H-C-β-PSO	1,113e3	5,091e3	4,516e4	2,855e4	4,0187e5

Tabela 16: Número de FLOPs para a função Griewank considerando o aumento das dimensões.

Metaheurísticas	FLOPs				
	$d = 2$	$d = 5$	$d = 10$	$d = 20$	$d = 50$
PSO	3,517e3	8,997e3	2,007e5	6,0122e5	2,925e6
β-PSO	1,999e3	6,181e3	7,189e4	4,139e5	2,001e6
C-PSO	2,412e3	7,101e3	9,994e4	5,005e5	2,376e6
H-C-β-PSO	8,812e2	4,111e3	2,134e4	8,855e4	7,0110e5

Tabela 17: Número de FLOPs para a função Rastrigin considerando o aumento das dimensões.

Metaheurísticas	FLOPs				
	$d = 2$	$d = 5$	$d = 10$	$d = 20$	$d = 50$
PSO	1,832e3	7,198e3	7,985e4	4,775e5	2,187e6
β-PSO	1,013e3	5,080e3	6,001e4	3,339e5	1,260e6
C-PSO	1,312e3	6,198e3	6,714e4	4,005e5	1,786e6
H-C-β-PSO	8,902e2	2,899e3	3,564e4	9,515e4	8,501e5

Tabela 18: Número de FLOPs para a função Rosenbrock considerando o aumento das dimensões.

Metaheurísticas	FLOPs				
	$d = 2$	$d = 5$	$d = 10$	$d = 20$	$d = 50$
PSO	2,012e3	5,981e3	8,091e4	6,101e5	2,313e6
β-PSO	1,787e3	5,711e3	6,595e4	4,321e5	1,569e6
C-PSO	1,992e3	6,108e3	7,910e4	5,817e5	2,138e6
H-C-β-PSO	1,108e3	3,199e3	4,104e4	8,799e4	9,154e5

para pequenas dimensões, a metaheurística C-PSO possuiu complexidade maior que o PSO tradicional, e isso se deve ao processo de geração dos mapas caóticos. Assim, para as funções benchmark avaliadas, a metaheurística H-C- β -PSO se mostrou como a mais adequada, pois a complexidade computacional foi menor em todos os cenários avaliados, inclusive com o aumento de dimensões.

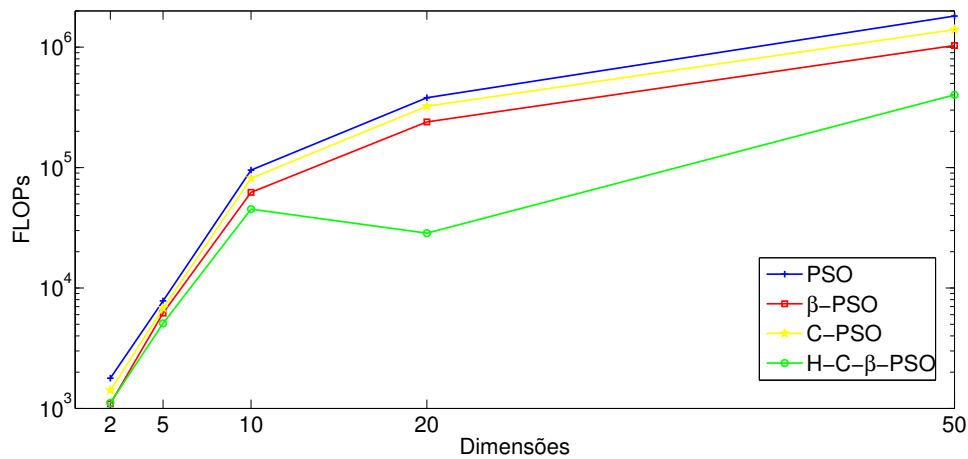


Figura 36: Número de FLOPs necessários para a convergência das metaheurísticas com a função Ackley considerando diferentes dimensões d .

Fonte: Autoria própria

Para trazer mais conhecimento sobre a complexidade das metaheurísticas utilizadas, principalmente em relação a abordagem proposta H-C- β -PSO é realizado um resumo da ro-

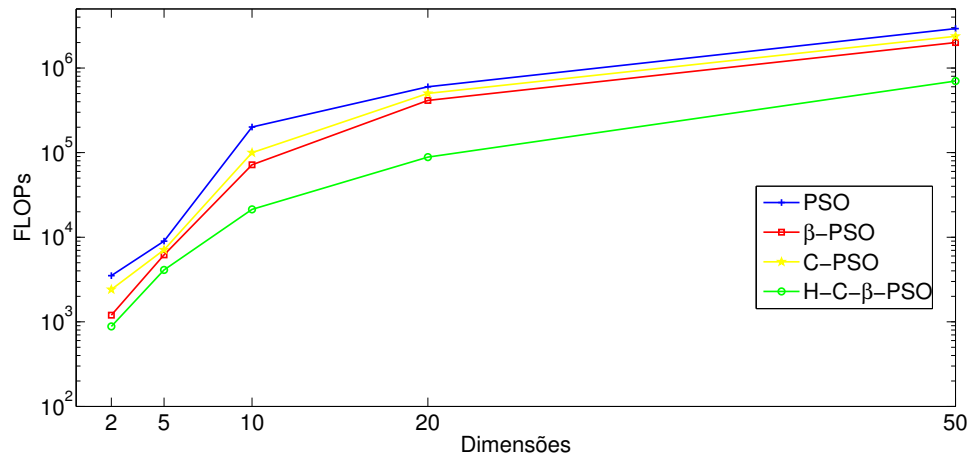


Figura 37: Número de FLOPs necessários para a convergência das metaheurísticas com a função Griewank considerando diferentes dimensões.

Fonte: Autoria própria

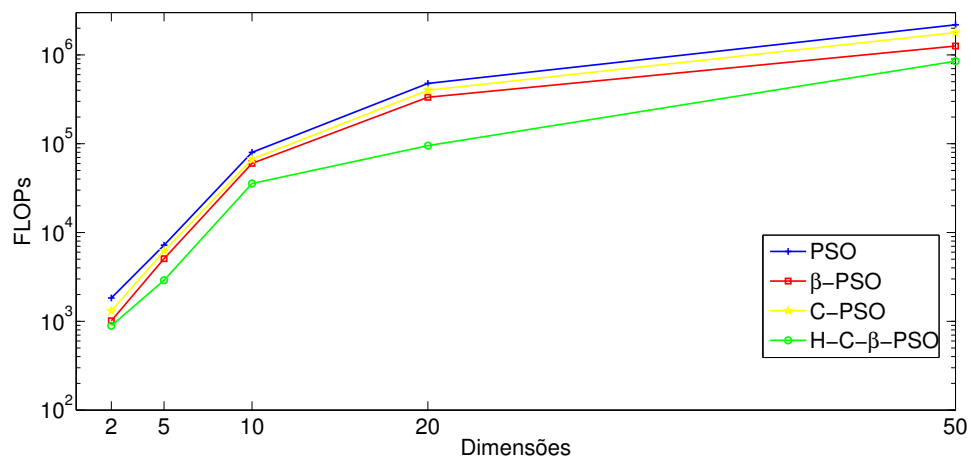


Figura 38: Número de FLOPs necessários para a convergência das metaheurísticas com a função Rastrigin considerando diferentes dimensões.

Fonte: Autoria própria

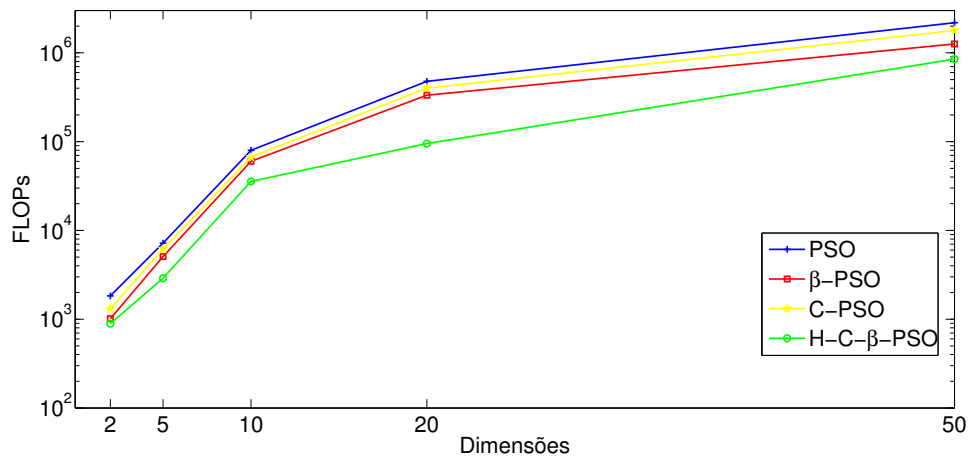


Figura 39: Número de FLOPs necessários para a convergência das metaheurísticas com a função Rosenbrock considerando diferentes dimensões.

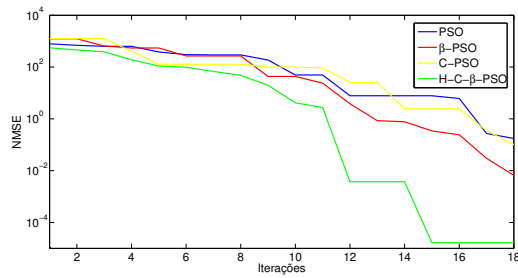
Fonte: Autoria própria

bustez desses algoritmos, de acordo com a Tabela 19, considerando para todos os cenários 50 realizações. Percebeu-se que todas as metaheurísticas obtiveram o máximo de robustez para todas as funções Benchmark com pequenas dimensões ($d = 2$, $d = 5$ e $d = 10$). Com o aumento de dimensões, principalmente para as funções Rastrigin e Rosenbrock ($d = 20$ e $d = 50$), nota-se que as metaheurísticas PSO tradicional, β -PSO e C-PSO não obtiveram 100% de robustez, já que não alcançaram convergência em todas as execuções realizadas. De maneira geral, a metaheurística proposta apresentou melhor robustez dentre os algoritmos, ainda que não conseguindo robustez máxima para todos os cenários, como por exemplo, para as funções Rastrigin e Rosenbrock com 50 dimensões.

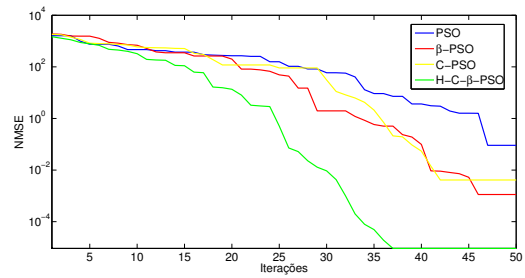
A qualidade das soluções obtidas pelas metaheurísticas foi avaliada através da métrica NMSE apresentada nas Figuras 40, 41, 42 e 43, que mostram a análise da evolução do NMSE em relação ao número de iterações para os cenários considerados de todas as funções Benchmark avaliadas. Considerando todas as funções e cenários, a metaheurística proposta H-C- β -PSO foi a que obteve o menor NMSE dentre todos os algoritmos analisados neste trabalho.

3.8 CONCLUSÕES DO CAPÍTULO

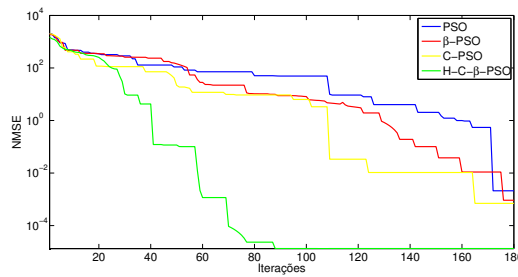
Neste Capítulo, foram apresentadas as metaheurísticas utilizadas por este trabalho, especialmente a metaheurística proposta Halton-Chaos- β -PSO, assim como foi realizada a validação desses algoritmos utilizando funções Benchmark e comparando o compromisso en-



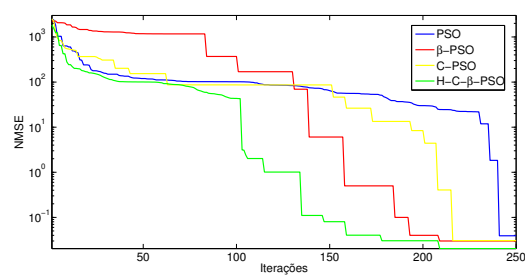
a) NMSE - Função Ackley 2-D



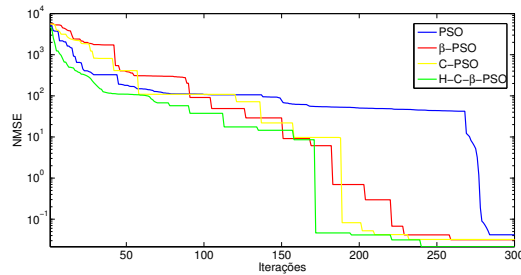
b) NMSE - Função Ackley 5-D



c) NMSE - Função Ackley 10-D



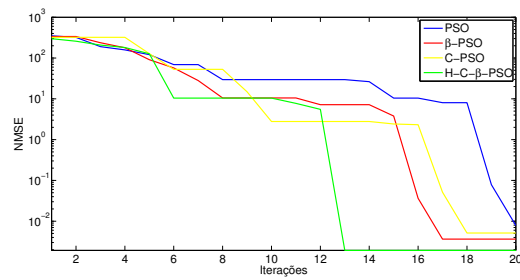
d) NMSE - Função Ackley 20-D



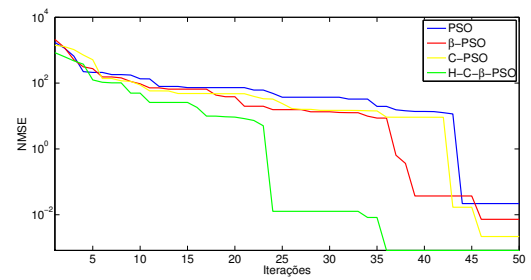
e) NMSE - Função Ackley 50-D

Figura 40: Número de iterações necessárias para a convergência das metaheurísticas com a função Ackley.

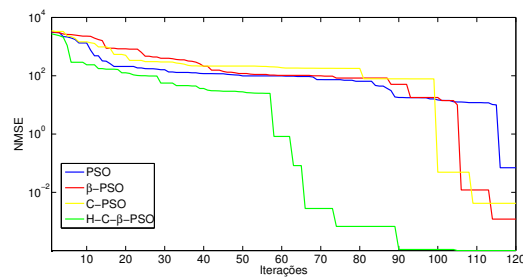
Fonte: Autoria própria



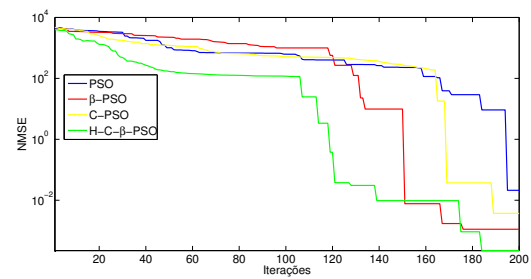
a) NMSE - Função Griewank 2-D



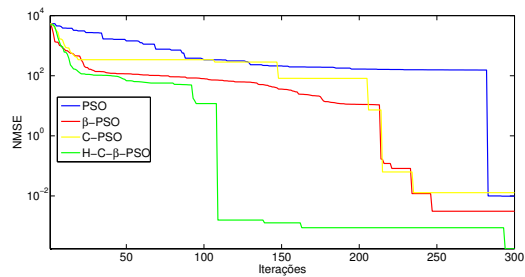
b) NMSE - Função Griewank 5-D



c) NMSE - Função Griewank 10-D



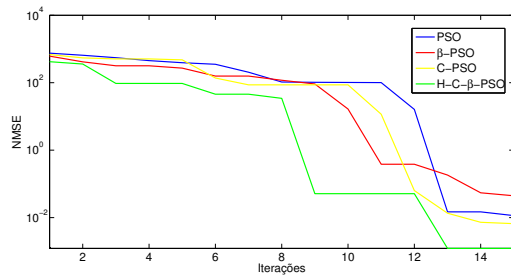
d) NMSE - Função Griewank 20-D



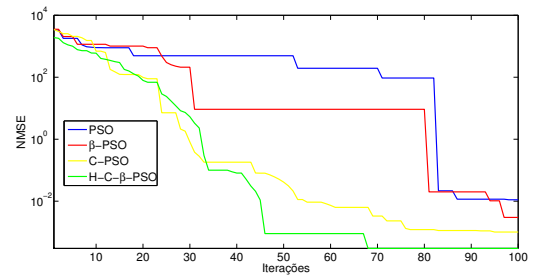
e) NMSE - Função Griewank 50-D

Figura 41: Número de iterações necessárias para a convergência das metaheurísticas com a função Griewank.

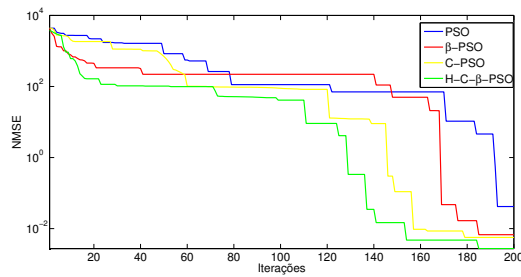
Fonte: Autoria própria



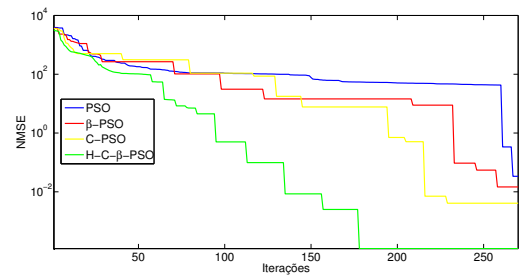
a) NMSE - Função Rastrigin 2-D



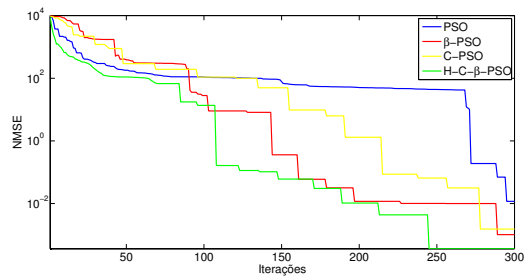
b) NMSE - Função Rastrigin 5-D



c) NMSE - Função Rastrigin 10-D



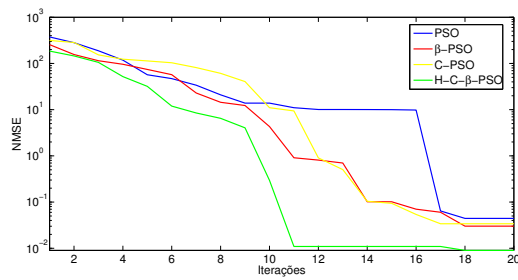
d) NMSE - Função Rastrigin 20-D



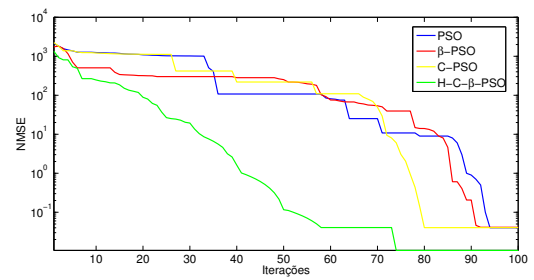
e) NMSE - Função Rastrigin 50-D

Figura 42: Número de iterações necessárias para a convergência das metaheurísticas com a função Rastrigin.

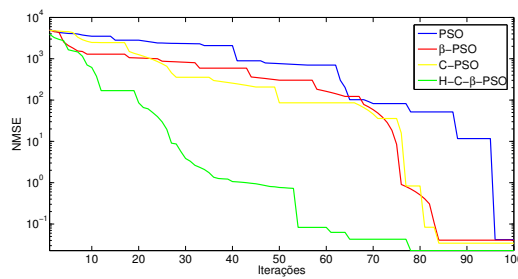
Fonte: Autoria própria



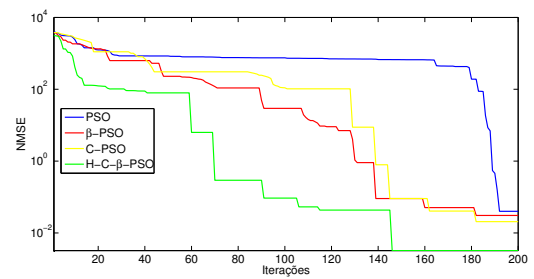
a) NMSE - Função Rosenbrock 2-D



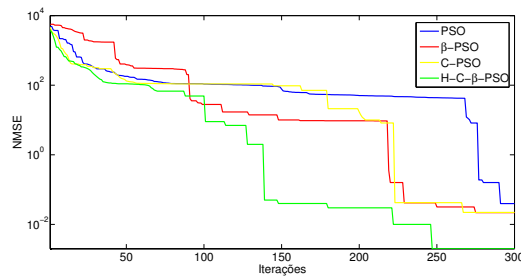
b) NMSE - Função Rosenbrock 5-D



c) NMSE - Função Rosenbrock 10-D



d) NMSE - Função Rosenbrock 20-D



e) NMSE - Função Rosenbrock 50-D

Figura 43: Número de iterações necessárias para a convergência das metaheurísticas com a função Rosenbrock.

Fonte: Autoria própria

Tabela 19: Robustez das metaheurísticas PSO, β -PSO, C-PSO e H-C- β -PSO para as funções Benchmarks.

Heurísticas	Benchmarks	Robustez				
		$d = 2$	$d = 5$	$d = 10$	$d = 20$	$d = 50$
PSO	Ackley	100%	100%	100%	98%	98%
	Griewank	100%	100%	98%	96%	94%
	Rastrigin	100%	100%	96%	96%	94%
	Rosenbrock	100%	100%	94%	92%	90%
β -PSO	Ackley	100%	100%	100%	100%	100%
	Griewank	100%	100%	100%	100%	100%
	Rastrigin	100%	100%	100%	98%	96%
	Rosenbrock	100%	100%	100%	96%	94%
C-PSO	Ackley	100%	100%	100%	100%	100%
	Griewank	100%	100%	100%	100%	98%
	Rastrigin	100%	100%	100%	98%	96%
	Rosenbrock	100%	100%	100%	96%	96%
HC-β-PSO	Ackley	100%	100%	100%	100%	100%
	Griewank	100%	100%	100%	100%	98%
	Rastrigin	100%	100%	100%	100%	98%
	Rosenbrock	100%	100%	100%	100%	98%

tre desempenho e complexidade das metaheurísticas com o PSO tradicional. Para todos os cenários e funções avaliadas, a abordagem proposta obteve melhor desempenho.

4 METAHEURÍSTICA HALTON-CHAOS- β -PSO PARA ESTIMAÇÃO DE PARÂMETROS DE UM MOTOR DE INDUÇÃO TRIFÁSICO

Máquinas de corrente alternada, incluindo Motores de Indução Trifásicos (MITs), possuem grandes vantagens, especialmente relacionadas à sua simples construção, quando comparadas a máquinas de corrente contínua (ONG et al., 1998). Mundialmente, os MITs são os motores mais utilizados devido à sua simples construção e confiabilidade (FITZGERALD et al., 2003).

Na maioria das aplicações dos MITs, é necessário o conhecimento prévio dos parâmetros de tais máquinas e esses parâmetros podem afetar as características de operação e controle. Geralmente, os parâmetros são fornecidos pelos fabricantes ou estimados por meio de testes tradicionais (URSEM; VADSTRUP, 2004). Entretanto, os fabricantes podem não fornecer todas as informações necessárias e os testes tradicionais podem não ser adequados, já que o modelo matemático do MIT possui características não-lineares e várias variáveis de estado que não podem ser medidas diretamente, como por exemplo, a indutância do estator (URSEM; VADSTRUP, 2004). Nesse contexto, técnicas não-lineares são aplicadas para a estimação e identificação de parâmetros para atender a essa demanda (PRAWIN et al., 2016). Ademais, a computação inteligente representa uma técnica adequada para a identificação e estimação de parâmetros dos MITs (URSEM; VADSTRUP, 2004); (ZHENG; LIAO, 2016); (BHOWMICK et al., 2016a), sendo que atualmente, técnicas metaheurísticas de otimização tem sido empregadas para a estimação de parâmetros de máquinas elétricas (SZCZEPANIAK; RUDNICKI, 1999).

Este Capítulo apresenta a aplicação da metaheurística H-C- β -PSO, baseada no algoritmo PSO tradicional para resolver o problema de estimação de parâmetros de um MIT sujeito à mudança de parâmetros durante sua operação, considerando o compromisso entre desempenho-complexidade do algoritmo. Para melhor avaliação da metaheurística proposta, também são aplicadas para a resolução do problema de estimação, uma heurística analítica-iterativa que combina o método Lagrangiano Aumentado (ALM) e o PSO, denominada ALPSO, um método ALM combinado com um método quase-Newton, e uma variante

híbrida do PSO com o algoritmo de busca gravitacional (PSOGSA). A eficiência e eficácia dos métodos avaliados são corroboradas por extensos resultados de simulação numérica, considerando o ajuste dos parâmetros de entrada dos algoritmos metaheurísticos. Os resultados revelaram uma redução notável nos valores de estimativa de erro obtidos pela metaheurística proposta, principalmente quando os métodos são implementados dinamicamente em aplicações em tempo real.

4.1 IDENTIFICAÇÃO DE PARÂMETROS DO MIT

Neste trabalho, os parâmetros estimados do MIT são: indutância do estator, resistência do estator e momento de inércia. O processo de estimação de parâmetros é realizado utilizando um motor gaiola de esquilo, em estado estacionário, na referência $dq0$ e negligenciando as perdas no entreferro. O modelo detalhado de operação desse motor pode ser encontrado em (KRISHNAN, 2001); (LEVI et al., 2007); (KRAUSE, 1986). As seções seguintes apresentam a modelagem e a formulação do problema para a estimação dos parâmetros do MIT.

4.1.1 MODELO DO MIT

Para este trabalho, o MIT é modelado na referência de estado estacionário $dq0$ (KRAUSE, 1986), sendo ilustrado na Figura 44.

As equações das tensões do estator e do rotor são dadas por (KRAUSE, 1986):

$$V_{sd} = R_s i_{sd} + \frac{d}{dt} \lambda_{sd} - \omega_d \lambda_{sq} \quad (29)$$

$$V_{sq} = R_s i_{sq} + \frac{d}{dt} \lambda_{sq} - \omega_d \lambda_{sd} \quad (30)$$

$$V_{rd} = R_r i_{rd} + \frac{d}{dt} \lambda_{rd} - \omega_m \lambda_{rq} \quad (31)$$

$$V_{rq} = R_r i_{rq} + \frac{d}{dt} \lambda_{rq} - \omega_m \lambda_{rd} \quad (32)$$

onde V_{sd} , V_{sq} , V_{rd} e V_{rq} representam as tensões do estator e do rotor nos eixos diretos e em quadratura, respectivamente (KRAUSE, 1986), i_{sd} , i_{sq} , i_{rd} e i_{rq} são as correntes do estator e do rotor nos eixos d e q , respectivamente. Os parâmetros λ_{sd} , λ_{sq} , λ_{rq} , e λ_{rd} representam o vetor do fluxo magnético do estator e do rotor na referência dq , respectivamente; R_s é a resistência do estator, R_r é a resistência do rotor; ω_d e ω_m representam a velocidade do sistema e a velocidade angular, respectivamente. As relações entre fluxo e correntes são

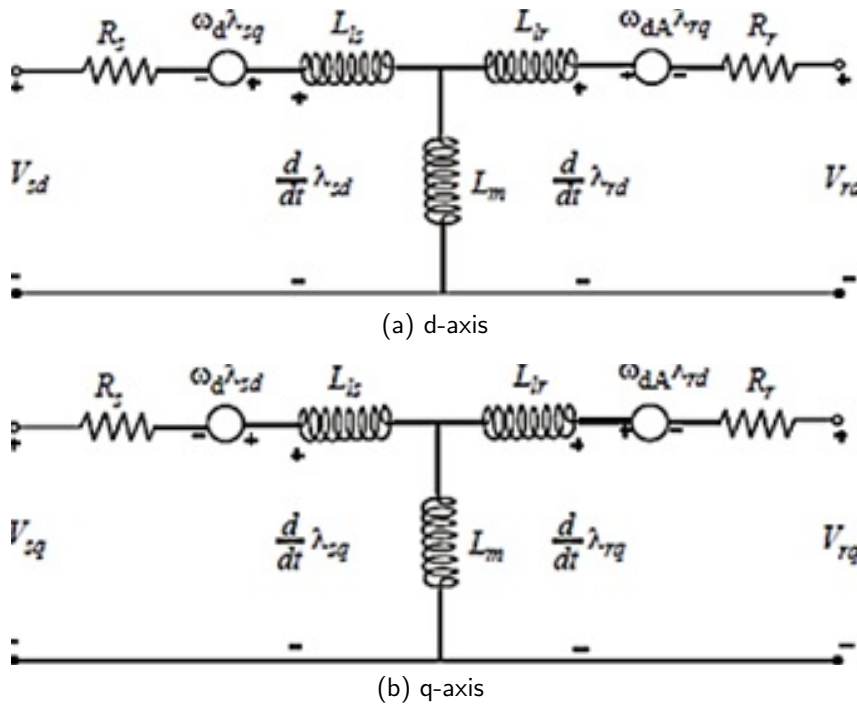


Figura 44: Circuito Equivalente do MIT na referência $dq0$

Fonte: (KRAUSE, 1986)

calculadas por:

$$\begin{bmatrix} \lambda_{sd} \\ \lambda_{sq} \\ \lambda_{rd} \\ \lambda_{rq} \end{bmatrix} = M \begin{bmatrix} i_{sd} \\ i_{sq} \\ i_{rd} \\ i_{rq} \end{bmatrix}; M = \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & L_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix} \quad (35)$$

onde, L_s , L_m , e L_r são a indutância do estator, a indutância de magnetização e a indutância do rotor, respectivamente. Assim, a parte elétrica do MIT pode ser descrita pela combinação das equações anteriores, considerando s como operador de Laplace, por:

$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \begin{bmatrix} R_s + L_s & -\omega_s L_s & sL_m & -\omega_s L_m \\ -\omega_s L_s & R_s + L_s & -\omega_s L_m & pL_m \\ sL_m & (\omega_s - \omega_r)L_m & R_r + sL_r & (\omega_s - \omega_r)L_r \\ -(\omega_s - \omega_r)L_m & sL_m & -(\omega_s - \omega_r)L_r & R_r + sL_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad (37)$$

Por superposição, isto é, adicionando os torques que atuam no eixo d e no eixo q dos enrolamentos do rotor, o torque instantâneo produzido na interação eletromecânica é dado por:

$$T_e = \frac{P}{2} \frac{3}{2} (\lambda_{rq} i_{rd} - \lambda_{rd} i_{rq}) \quad (38)$$

O torque eletromagnético pode ser expresso em termos de indutância por:

$$T_e = \frac{P}{2} \frac{3}{2} L_m (i_{sq} i_{rd} - i_{sd} i_{rq}) \quad (39)$$

A parte mecânica do motor é modelado de acordo com:

$$\frac{d}{dt} \omega_m = \frac{T_e - T_L}{J} \quad (40)$$

onde, ω_m é a velocidade angular, T_e é o torque eletromagnético, T_L é o torque de carga, J é o momento de inércia. Um conjunto de tensões de entrada é aplicado ao enrolamento do estator. As equações anteriores demonstram o modelo dinâmico do MIT, que é altamente não-linear. As variáveis de estado são representadas pelas resistências do estator e do rotor, indutâncias do rotor e do estator e a indutância mútua. Assim, o modelo de estados é calculado por:

$$x = [\lambda_{qs} \quad \lambda_{ds} \quad \lambda_{qr} \quad \lambda_{dr} \quad \omega_d] \quad (41)$$

$$\Theta = [V_{qs} \quad V_{ds} \quad V_{qr} \quad V_{dr}] \quad (42)$$

$$y = [i_{qs} \quad i_{ds}] \quad (43)$$

4.1.2 PROBLEMA DE ESTIMAÇÃO DE PARÂMETROS DO MIT

Uma abordagem baseada em modelos depende de uma formulação matemática explícita do sistema monitorado e o principal objetivo é garantir a consistência entre as medidas do sistema e os valores estimados baseado em um modelo analítico (NIKLANJBAR et al., 2009). Assim, modelos paramétricos podem ser expressos por equações diferenciais ordinárias e parciais. Considerando $u(t)$ como entradas e $y(t)$ como saídas do sistema, onde t representa o tempo discreto, a dinâmica fundamental do sistema pode ser descrita por:

$$\dot{x} = F(x, \Theta, u) \quad (45)$$

$$y = G(x, \Theta) \quad (46)$$

onde $\dot{x}$ é o vetor de estados e Θ é o vetor de parâmetros a ser identificado.

Para a estimação de parâmetros, a ideia básica consiste em comparar as variáveis de saída do sistema com as saídas do modelo utilizando um método de otimização baseado em uma função custo, que é definida como sendo a medida que indica o quão próximas as saídas do modelo estão em relação às saídas do sistema. A Figura 45 resume o processo de estimação

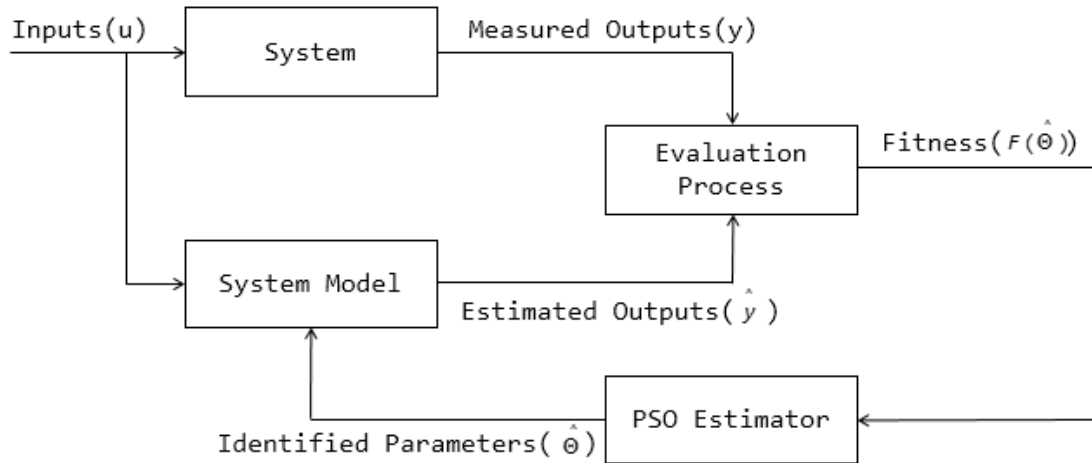


Figura 45: Processo de Estimação de Parâmetros.

Fonte: Autoria própria

de parâmetros utilizado neste trabalho. O vetor do modelo do sistema $\hat{\Theta}$ para a estimação de parâmetros é definido como:

$$\hat{x} = F(\hat{x}, \hat{\Theta}, u) \quad (48)$$

$$\hat{y} = G(\hat{x}, \hat{\Theta}) \quad (49)$$

onde a resistência do estator R_s , a indutância do estator L_s e o momento de inércia J são os parâmetros medidos na saída do sistema, enquanto $\hat{\Theta} = [\hat{R}_s, \hat{L}_s, \hat{J}]$ são os parâmetros estimados e que serão identificados pelo processo de otimização. No processo de estimação de parâmetros, as mesmas entradas u são aplicadas ao sistema e ao modelo matemático. Os parâmetros estimados $\hat{x}$ e $\hat{y}$ representam as saídas e os estados do modelo.

Para a avaliação dos parâmetros estimados, as saídas do sistema y são comparadas com as saídas do modelo $\hat{y}$. Dessa maneira, a função custo pode ser definida utilizando o erro quadrático, calculado por:

$$F(\hat{\Theta}) = \int_0^t (y - \hat{y})^2 dt \quad (51)$$

onde F é igual a zero quando $\hat{\Theta} = \Theta$. Assim, a função custo pode ser utilizada para guiar o processo de busca em direção a um melhor processo de estimação de parâmetros. Convertendo o problema de estimação em um problema de otimização, tem-se:

$$\begin{aligned} \arg \min \quad & F(\hat{\Theta}) \\ \text{s.a.} \quad & \hat{\Theta}_{\min} < \hat{\Theta} < \hat{\Theta}_{\max} \end{aligned} \quad (52)$$

4.2 METAHEURISTICAS PARA ESTIMAÇÃO DE PARÂMETROS DO MIT

Nesta seção, a habilidade do algoritmo PSO é comparada com algumas de suas variantes, especialmente com a metaheurística Halton-Chaos- β -PSO, para resolver o problema de estimação de parâmetros de um motor de indução trifásico. A seguir, os métodos ALM, ALPSO, PSOGSA são explicados. As metaheurísticas PSO, β -PSO, C-PSO e a abordagem proposta H-C- β -PSO estão descritas no Capítulo 4.

4.2.1 ALGORITMOS ALM E ALPSO

O método de otimização Lagrangiano Aumentado busca resolver um problema de programação não-linear restrito, resolvendo uma sequência de problemas mais diretamente. Conforme descrito em Martinez et al. (2020) pode-se combinar uma técnica analítico-iterativa (ALM) com um método do tipo quase Newton ou com um método heurístico evolutivo (PSO) visando melhorar o desempenho do método de otimização híbrido. Portanto, em uma combinação, cada subproblema de ALM pode ser resolvido aproximadamente pelo PSO ou pelo BFGS, Broyden, Fletcher, Goldfarb e Shanno (MARTINEZ et al., 2020). Além disso, cada problema considera restrições simples definidas por uma caixa, $\tilde{\Theta}_{\min} < \hat{\Theta} < \tilde{\Theta}_{\max}$. Assim, cada problema considera restrições simples definidas por uma caixa, $\tilde{\Theta}_{\min} < \hat{\Theta} < \tilde{\Theta}_{\max}$. Portanto, o espaço de busca para o problema de estimação de parâmetros do MIT pode definido como:

$$\tilde{\Theta}_{\min} = \frac{1}{\zeta} \cdot [R_s, L_s, J] \quad \text{e} \quad \tilde{\Theta}_{\max} = \zeta \cdot [R_s, L_s, J] \quad (54)$$

sendo $\zeta > 1$.

Sob este conjunto, o PSO ou alternativamente o BFGS minimiza a função custo. Para o problema de estimação definido em 73, considera-se a função Lagrangiana como função custo para o PSO ou BFGS:

$$\mathcal{A}_\rho(\Theta, \mu) = F(\hat{\Theta}) + \frac{\rho}{2} \left[[F(\hat{\Theta}) - m_k]^+ + \frac{\mu}{\rho} \right]^2 \quad (55)$$

no qual $[\alpha]^+ = \max[0, \alpha]$; o parâmetro de penalidade $\rho > 0$ e $\mu \geq 0$ são aproximações dos multiplicadores de Lagrange. O fator $[F(\hat{\Theta}) - m_k]^+$ é incluído com o objetivo de se usar a estrutura ALM para acelerar a minimização ou maximização da função custo $F(\hat{\Theta})$, e a constante m_k representa um valor menor ou igual ao valor F na iteração anterior do ALM. Informações detalhadas sobre o ALM são encontradas em ((??)).

O algoritmo ALPSO é baseado no PSO tradicional com atualização periódica dos

multiplicadores de Lagrange do parâmetro de penalidade para a função custo. O algoritmo ALM consiste no método Lagrangiano Aumentado com o método Quasi-Newton para resolver sub-problemas sequencialmente.

4.2.2 ALGORITMO PSOGSA

Hibridização de algoritmos pode ser considerada uma estratégia adequada para superar certos problemas existentes em diferentes áreas da inteligência computacional. Dois algoritmos podem ser hibridizados com baixo ou alto nível (TALBI, 2002). A estratégia PSOGSA implantada neste trabalho representa uma hibridização coevolucionária de baixo nível, já que o PSO tradicional e o algoritmo GSA são implementados em paralelo. O algoritmo de busca gravitacional (GSA), que emprega N agentes, inicia posicionando esses N agentes no espaço de busca. Ao longo das iterações, a força gravitacional do agente j sob o agente i no tempo específico k , chamado de $F_{ij}^d(k)$, é definido por (RASHEDI et al., 2009):

$$F_{ij}^d(k) = G(k) \frac{M_{pi}(k) \times M_{aj}(k)}{R_{ij}(k) + \bar{\varepsilon}} [x_j^d(k) - x_i^d(k)], \quad (56)$$

no qual M_{aj} é a massa ativa gravitacional relacionado ao agente j , M_{pi} é a massa gravitacional passiva relacionado ao agente i , $G(k)$ é a constante gravitacional no tempo k , $\bar{\varepsilon}$ é uma constante e $R_{ij}(k)$ é a distância euclidiana entre dois agentes i e j , $x_i^d(k)$ é a posição da partícula i na iteração k , $x_j^d(k)$ é a posição da partícula j na iteração k ; e a função $G(k)$ é calculada por:

$$G(k) = G_0 \exp \left[\frac{-\alpha \cdot k}{\mathcal{I}_{\max}} \right], \quad (57)$$

sendo α e G_0 os coeficientes descendentes e o valor inicial, respetivamente, k é a iteração atual, e $\mathcal{I}_{\max}$ é o número máximo de iterações. Em um problema com espaço de d dimensões diferentes, a força total agindo sob o agente i é calculada por:

$$F_i^d(k) = \sum_{j=1, j \neq i}^N \text{rand}_j \cdot F_{ij}^d(k) \quad (58)$$

sendo $\text{rand}_j \sim \mathcal{U}[0, 1]$ um valor aleatório com distribuição uniforme no intervalo $[0, 1]$.

De acordo com a lei do movimento, a aceleração do agente é proporcional à força resultante e o inverso de sua massa. Assim, a aceleração $ac_i^d(k)$ do agente i na iteração k é calculada por:

$$ac_i^d(k) = \frac{F_i^d(k)}{M_i(k)}, \quad (59)$$

no qual M_i é a massa do i -ésimo objeto. Assim, a atualização da velocidade e da posição do

i -ésimo agente na $(k + 1)$ -ésima iteração são dadas, respectivamente, por:

$$v_i^d(k + 1) = \text{rand} \cdot v_i^d(k) + \text{ac}_i^d(k) \quad (60)$$

$$x_i^d(k + 1) = x_i^d(k) + v_i^d(k + 1) \quad (61)$$

Proposto em (MIRJALILI; HASHIM, 2010), a hibridização do PSO com o GSA possui um baixo nível co-evolucionário de hibridização. A ideia básica do PSOGA é combinar a capacidade social do PSO com a capacidade de busca local do GSA. Dessa maneira, a atualização de velocidade do agente i na iteração $k + 1$ é dada por:

$$v_i(k + 1) = w' \cdot v_i(k) + c'_1 \cdot \text{rand} \cdot \text{ac}_i(k) + c'_2 \cdot \text{rand} \cdot [g^* - x_i^d(k)] \quad (63)$$

sendo c'_j o fator de peso, w' é uma função de peso, o número aleatório $\text{rand} \sim \mathcal{U}[0, 1]$; a aceleração do agente i na iteração k é $\text{ac}_i(k)$ e g^* é a melhor posição encontrada pelo enxame na iteração k . Assim, em cada iteração, as posições das partículas no algoritmo PSOGA é atualizada de acordo com:

$$x_i(k + 1) = x_i(k) + v_i(k + 1). \quad (64)$$

Neste algoritmo híbrido, todas as partículas são inicializadas aleatoriamente, enquanto cada partícula é considerada uma potencial solução. Após a inicialização, a força gravitacional, a constante gravitacional e o as forças resultantes entre as partículas são calculadas usando (56), (57) e (58), respectivamente. A cada iteração, a melhor solução é atualizada. Depois do cálculo das acelerações com os valores das melhores soluções, a velocidade de todos s agentes é calculada por (63). Finalmente, as posições dos agentes são definidas por (64). O processo de atualização da velocidade e posição é finalizado quando o critério de parada é encontrado.

4.3 CALIBRAÇÃO DOS PARÂMETROS DE ENTRADA

O programa MATLAB-Simulink foi usado para as simulações numéricas do problema de estimação de parâmetros do MIT. Os principais resultados das simulações são analisados nesta seção considerando os parâmetros do MIT, como apresentado na Tabela 20.

Para analisar o comportamento e a capacidade de estimação dos algoritmos, os parâmetros estimados do MIT ($p_{\text{plant}} = [J, R_s, L_s]$) foram sujeitos a uma mudança durante a execução (aproximadamente na iteração 40). Assim, após a mudança, os novos valores dos parâmetros são:

$$p_{\text{plant}} = [J \cdot 0,04; R_s \cdot 1,6725; L_s \cdot 0,008961]$$

Tabela 20: Parâmetros do MIT.

Parâmetro	Valor	Unidade
Potência (P)	5	[hp]
Tensão (V)	460	[V]
Frequência Nominal (f)	60	[Hz]
Resistência do Estator (R_s)	1,115	[Ω]
Indutância do Estator (L_s)	0,005974	[H]
Resistência do Rotor (R_r)	1,083	[Ω]
Indutância do Rotor (L_r)	0,005974	[H]
Momento de Inércia (J)	0,02	[(kg.m) ²]
Número de Polos	4	

Nesta seção e nos resultados numéricos, o espaço de busca para o problema de estimação é definido como:

$$\tilde{\Theta}_{\min} = \frac{1}{\zeta} \cdot [R_s, L_s, J] \quad \text{e} \quad \tilde{\Theta}_{\max} = \zeta \cdot [R_s, L_s, J]$$

no qual foi adotado $\zeta = 5$.

4.3.1 CALIBRAÇÃO DOS PARÂMETROS DE ENTRADA – PSO TRADICIONAL

Considerando os valores iniciais do MIT como ($p_{\text{plant}} = [0,02, 1,115, 0,005974]$), que representam o momento inercial do rotor, a resistência do estator e a indutância do estator, os valores dos parâmetros de entrada do algoritmo PSO tradicional empregados para todas as realizações são definidos na Tabela 21.

Tabela 21: Valores dos Parâmetros do PSO

Parâmetros	Valores
Número de partículas ($\mathcal{S}$)	15
Aceleração da partícula (c_1)	2,05
Aceleração global (c_2)	2,00
Peso da inércia (ω)	0,73
Máximo de iterações ($\mathcal{I}_{\max}$)	$\mathcal{I} = 80$
Velocidade máxima ($v_{\max}$)	p_{plant}
Velocidade mínima ($v_{\min}$)	$p_{\text{plant}}/300$

Adicionalmente, os resultados dos algoritmos β -PSO, C-PSO e a metaheurística proposta são comparados em termos do compromisso entre desempenho e complexidade para os mesmos parâmetros de entrada. Assim, esses parâmetros são utilizados para todas as metaheurísticas. Para o algoritmo β -PSO, os parâmetros p, q foram ajustados de acordo com a Distribuição Beta; para o algoritmo C-PSO, o fator de amortecimento foi cuidadosamente ajustado, assim como o parâmetro de combinação dos números primos para o algoritmo H-PSO.

Os resultados numéricos dos ajustes dos parâmetros de entrada são discutidos nas próximas seções.

4.3.2 CALIBRAÇÃO DOS PARÂMETROS DE ENTRADA – β -PSO

A Figura 46(a) ilustra o número de iterações necessárias para a convergência do algoritmo β -PSO para a estimação de parâmetros, empregando diferentes valores para o parâmetro p . Pode-se notar que o valor desse parâmetro pode influenciar na velocidade de convergência do algoritmo, já que existe uma variação no número de iterações necessárias, que oscilou entre $\mathcal{I} \approx 20$ ($p = 0,8$) e $\mathcal{I} \approx 50$ ($p = 2$), no intervalo adotado $p \in [0, 1, 2]$. Após exaustivos testes, o valor do parâmetro p foi fixado em $p = 0,8$, considerando a avaliação de uma grande quantidade de combinação dos parâmetros. A Figura 46(b) ilustra o número de iterações necessárias para a convergência do algoritmo β -PSO quando diferentes valores para o parâmetro q são adotados. Nessa avaliação, o valor de p foi fixado em $p = 0,8$. Assim, a Figura 46(b) ilustra o número de iterações necessárias para a convergência do algoritmo e pode-se notar que quando $q = 1,3$, o algoritmo obtém convergência mais rápida ($\mathcal{I} \approx 20$), no intervalo de $q \in [0.1; 2]$. Dessa maneira, os valores $p = 0,8$ e $q = 1,3$ representaram o melhor compromisso entre diversificação e exploração para o problema de estimação de parâmetros do MIT.

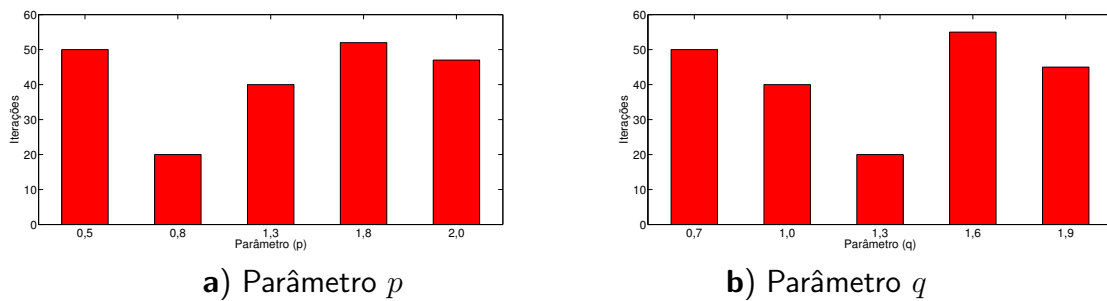


Figura 46: Número de iterações necessárias para a convergência do algoritmo β -PSO considerando diferentes valores para os parâmetros a) p e b) q .

Fonte: Autoria própria

4.3.3 CALIBRAÇÃO DE PARÂMETROS DE ENTRADA – CHAOS-PSO

Os métodos PSO tradicional e C-PSO para problemas de estimação apresentam vários parâmetros equivalentes; portanto, esses parâmetros de entrada utilizados para o PSO também foram utilizados para o algoritmo C-PSO. Por outro lado, para o C-PSO, o fator de amortecimento é gerado pelo mapa logístico com um parâmetro de controle para obter o melhor

comportamento caótico e convergência mais rápida, considerando o menor número de iterações (SANTANA et al., 2018).

A Figura 47 mostra o número de iterações necessárias para a convergência do algoritmo C-PSO quando diferentes valores de fatores de amortecimento são adotados, no intervalo $p \in [0, 1, 2]$. Pode-se notar que para $\xi = 3$ obteve-se o menor número de iterações para convergência ($\mathcal{I} \approx 20$); assim nos resultados numéricos este valor foi adotado.

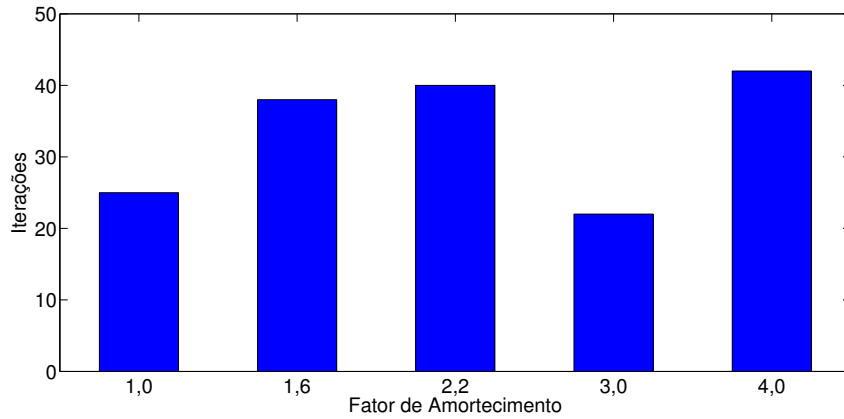


Figura 47: Número de iterações necessárias para a convergência do algoritmo C-PSO considerando diferentes valores de fator de amortecimento.

Fonte: Autoria própria

4.3.4 CALIBRAÇÃO DOS PARÂMETROS DE ENTRADA – HALTON-PSO

Neste trabalho, o operador σ_r foi gerado através da utilização do algoritmo B&W e para a escolha do melhor valor de $b_l(i)$ (DIGEHSARA et al., 2020) para o problema de estimação de parâmetros.

A Figura 48 ilustra o número de iterações necessárias para a convergência do algoritmo H-PSO quando diferentes valores de $b_l(i)$ são adotados. Pode-se observar que a variação do número de iterações para a convergência do algoritmo oscilou entre $\mathcal{I} \approx 20$ ($b = 11$) e $\mathcal{I} \approx 52$ ($b = 3$). Esse comportamento está associado à variação das permutações com a alteração do número primo como base. O valor $b = 11$ representou o melhor compromisso entre diversificação e intensificação para o problema de estimação de parâmetros usando o algoritmo H-PSO para a inicialização da população inicial.

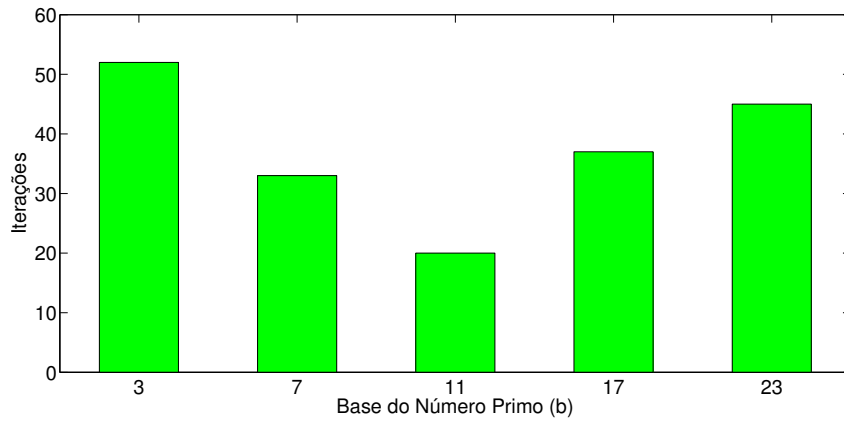


Figura 48: Número de iterações necessárias para a convergência do algoritmo H-PSO considerando diferentes valores de número primos para a base b .

Fonte: Autoria própria

4.3.5 CALIBRAÇÃO DOS PARÂMETROS DE ENTRADA DO ALGORITMO HÍBRIDO PSO-GSA

Para o problema de estimação de parâmetros, os valores comuns adotados para os parâmetros de entrada no PSO tradicional foram implantados, conforme definido na Tabela 21. Além disso, para a parte da pesquisa gravitacional, considerou-se a constante gravitacional como $G_0 = 1$.

4.4 COMPARAÇÃO ENTRE OS MÉTODOS DE ESTIMAÇÃO DE PARÂMETROS

Nesta seção, os algoritmos para estimação de parâmetros do MIT são avaliados. A metaheurística proposta Halton-Chaos- β -PSO é comparada com quatro algoritmos: a) PSO e duas variações; b) β -PSO; c) C-PSO; d) método híbrido ALPSO. Além disso, o problema de estimação é avaliado considerando as saídas do modelo do sistema. Os parâmetros do MIT estão descritos na Tabela 20. Os algoritmos são analisados em termos de precisão, isto é, qualidade de convergência (medida através do erro quadrático médio), complexidade computacional e compromisso entre desempenho-complexidade.

4.4.1 CRITÉRIO DE CONVERGÊNCIA

Neste trabalho, é usada a combinação dos seguintes critérios de convergência:

- a) Máximo de iterações $\mathcal{I}_{\max} = 80$;
- b) Erro quadrático médio (MSE) entre as saídas do modelo e dos parâmetros estimados pelos algoritmos após a t -ésima iteração, considerando 10^{-2} como o valor mínimo para convergência.

Considerando a otimização do problema de estimação dos parâmetros, a referência para a solução ótima é dada pela solução MSE. Neste trabalho, para uma comparação justa entre os algoritmos, um teste de convergência ($\mathcal{C}$) foi aplicado por todos os algoritmos considerando 100 % de sucesso quando a seguinte relação foi mantida:

$$\mathcal{C}[t] = (y[t] - \hat{y}[t])^2 \leq 10^{-2} \quad (65)$$

onde, $y[t]$ é o valor do parâmetro do modelo na t -ésima iteração e $\hat{y}[t]$ é o valor estimado do parâmetro obtido pelos algoritmos.

4.4.2 SIMULAÇÕES E RESULTADOS

As simulações foram realizadas com o MATLAB em um computador pessoal com 4GB de RAM, Intel Core i5 @ 1,6 GHz. Além disso, os desempenhos dos algoritmos de estimação de parâmetros β -PSO, Chaos-PSO e Halton-Chaos- β -PSO foram comparados ao algoritmo PSO tradicional, validado e comparado previamente em (BHOWMICK et al., 2017), (BHOWMICK et al., 2018) e (BHOWMICK et al., 2016b). A fim de aprofundar a análise das simulações e validação dos resultados, análises comparativas com os métodos ALM, ALPSO e PSOGSA foram incluídas nos testes. Considerando os parâmetros descritos anteriormente na Tabela 20 e a alteração dos parâmetros durante a operação do MIT, o desempenho dos algoritmos metaheurísticos (incluindo o método híbrido) e o método analítico ALM foram comparados em termos de parâmetros estimados, número de iterações, erro quadrático médio (MSE) e número de FLOPs.

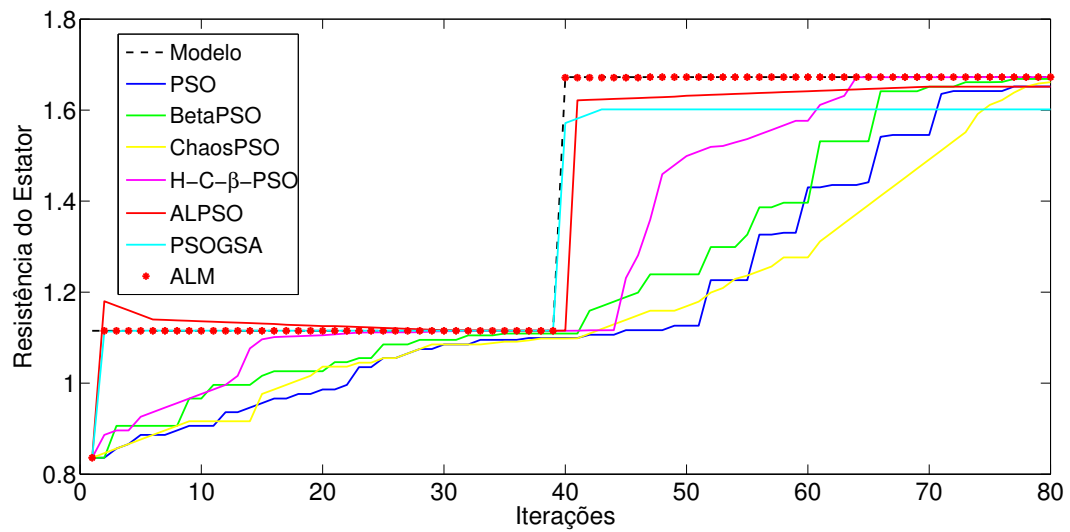
A Figura 49 apresenta os resultados numéricos para mais 50 realizações da estimação dos parâmetros para todos os algoritmos, incluindo resistência e indutância do estator e momento de inércia. De fato, a Figura 49(a) mostra a evolução da resistência do estator, indutância do estator e momento de inércia vs. o número de iterações obtidas pelos métodos analisados. A linha tracejada horizontal representa os valores dos parâmetros do modelo, obtidos através do procedimento matemático, e aqui, empregados para validar os parâmetros estimados pelos algoritmos. Conforme observado, antes da alteração dos parâmetros do modelo, o algoritmo HC- β -PSO pode atingir os valores dos parâmetros do modelo em torno de 30 iterações. Após a mudança dos valores dos parâmetros, a abordagem proposta encontrou os valores de referência do modelo por volta da 60-ésima iteração, permanecendo mais rápido que os outros métodos e demonstrando um escape distinto de ótimos locais, mesmo no cenário de mudança dos parâmetros. Além disso, observou-se que a evolução transiente do algoritmo foi mais adequada (comportamento suave, monotônico e velocidade de convergência).

Por outro lado, o método ALPSO alcançou a solução desejada na 50-ésima iteração após a alteração dos parâmetros, mas se afastou da solução após a 60-ésima iteração e retornou à mesma na 75-ésima iteração. O método ALM atingiu a melhor solução com apenas 5 iterações e este desempenho se deve ao fato do método já trabalhar com estimativa adequada do parâmetro e com uma boa aproximação para o multiplicador de Lagrange. Porém, vale ressaltar que o custo computacional desse método é alto e se concentra nas primeiras iterações até a obtenção da solução. O método híbrido PSOGSA teve um bom desempenho após a alteração dos parâmetros e abordou as soluções desejadas logo após a iteração 50.

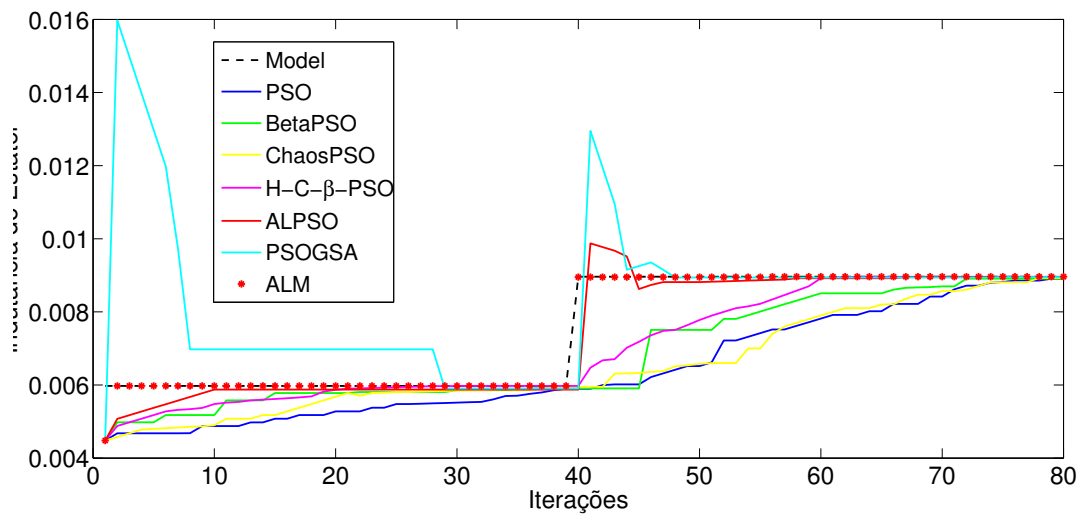
A Figura 50 ilustra as partículas de enxame no espaço de busca na iteração final do processo de estimação dos parâmetros do MIT para a) PSO, b) β -PSO, c) C-PSO, d) HC- β -PSO, com os respectivos valores de aptidão. Como se pode notar, a melhor partícula (marcador de ponto vermelho) para todos os métodos quase atingiu os melhores valores dos parâmetros, e as partículas do enxame foram distribuídas muito próximas do melhor valor global, enquanto a precisão admissível de adequação foi alcançada para todos os algoritmos. No entanto, as partículas do algoritmo $H-C-\beta$ -PSO, ilustradas na Figura 50(d), ficaram mais próximas do melhor global que as partículas do PSO e C-PSO, alcançando o melhor valor global que esses algoritmos, uma vez que a distribuição Beta permitiu o aumento da diversidade, auxiliando na exploração (diversificação) de regiões encobertas no espaço de busca durante o processo de estimação.

Além disso, na Figura 50 (c) pode-se observar o comportamento final das partículas da metaheurística C-PSO no espaço de busca e notar que o fator de amortecimento gerado por mapas caóticos pode ajudar o algoritmo a evitar convergência prematura, aumentando a diversificação sem perda da capacidade de exploração. Embora o melhor valor de fitness alcançado pelo C-PSO seja maior do que o alcançado pelo algoritmo β -PSO, este ainda obteve melhor desempenho que o algoritmo tradicional, além da obtenção de menor valor que a precisão admissível de fitness. Além disso, a Figura 50 (d) mostra a distribuição final das partículas para o algoritmo HC- β -PSO; neste caso, a melhor partícula atingiu a posição de valor global muito próxima (ótimo global), corroborando que as características agregadas podem melhorar a qualidade da convergência no processo de estimação dos parâmetros. Em conclusão, o algoritmo H-C- β -PSO, bem como os métodos analíticos atingiram os melhores valores de fitness (2,0173e-04 e 1,1250e-4, respectivamente), enquanto PSO, β -PSO, C-PSO, ALPSO e PSOGSA obtiveram valores muito maiores: (8,0580e-03, 1,1680e-03, 5,6280e-03, 8,1310e-4 e 7,3828e-4), respectivamente.

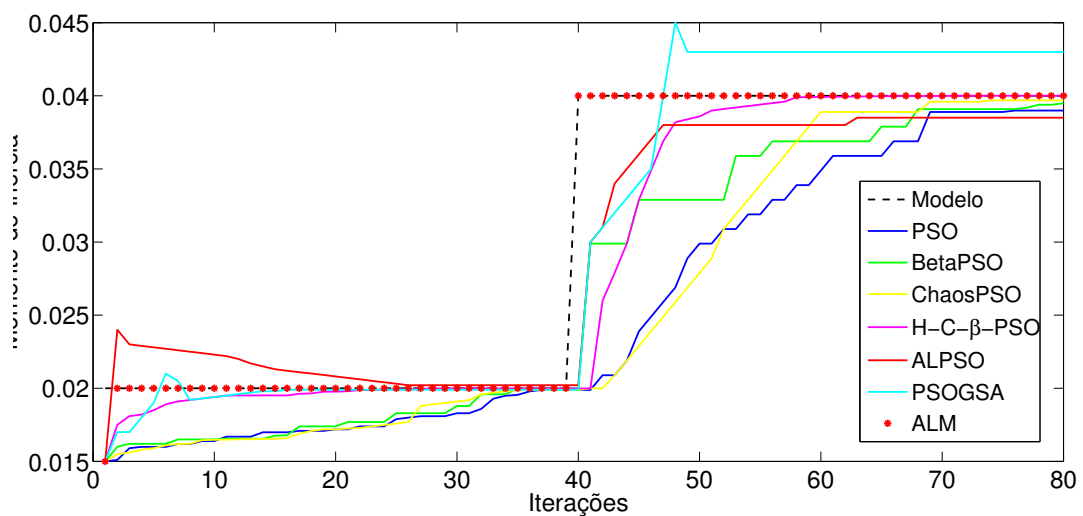
O erro quadrático médio (MSE) do parâmetro estimado igual a zero é ideal, mas não



(a) Resistência do Estator



(b) Indutância do Estator



(c) Momento de Inércia

Figura 49: Média de 50 realizações dos algoritmos para a estimação de parâmetros do MIT: a) Resistência do estator; b) Indutância do estator; c) Momento de Inércia.

Fonte: Autoria própria

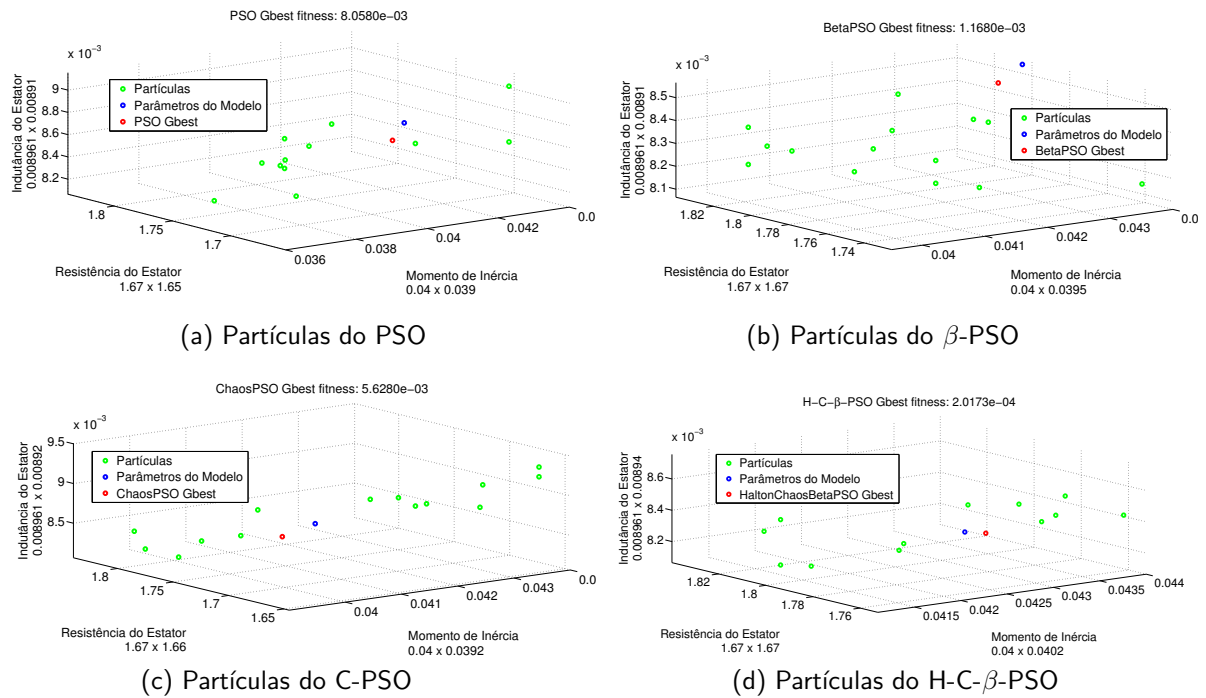


Figura 50: Posição final do enxame: a) PSO; b) BetaPSO; c) ChaosPSO; d) HaltonChaos-BetaPSO

Fonte: Autoria própria

é possível na maioria das situações. Assim, o MSE próximo de zero significa que o estimador prevê observações com grande precisão. Neste trabalho, o MSE é utilizado para indicar o quão bom é o estimador, conforme ilustrado na Figura 51, considerando o MSE a média de mais de 50 realizações. Considerando a alteração dos parâmetros do MIT em torno da 40-ésima iteração, o MSE tende a ficar mais alto após esta iteração. No entanto, todos os algoritmos atingiram a precisão mínima definida em torno da 65-ésima iteração. Observou-se que entre os métodos metaheurísticos evolutivos analisados, o algoritmo HC - β -PSO atingiu o menor MSE antes e depois da alteração dos parâmetros, demonstrando um desempenho promissor ao manter uma operação de convergência suave ao longo do processo de estimação.

4.4.3 COMPLEXIDADE COMPUTACIONAL

Esta seção avalia a complexidade de todos os algoritmos analisados e o número de FLOPs necessários para as operações principais, sendo resumidos na Tabela 22.

Em geral, o método HC - β -PSO proposto revelou quase o mesmo número de FLOPs que o PSO tradicional e β -PSO. Pode-se observar que a complexidade do HC - β -PSO em termos de FLOPs não apresentou um aumento considerável de complexidade em relação aos

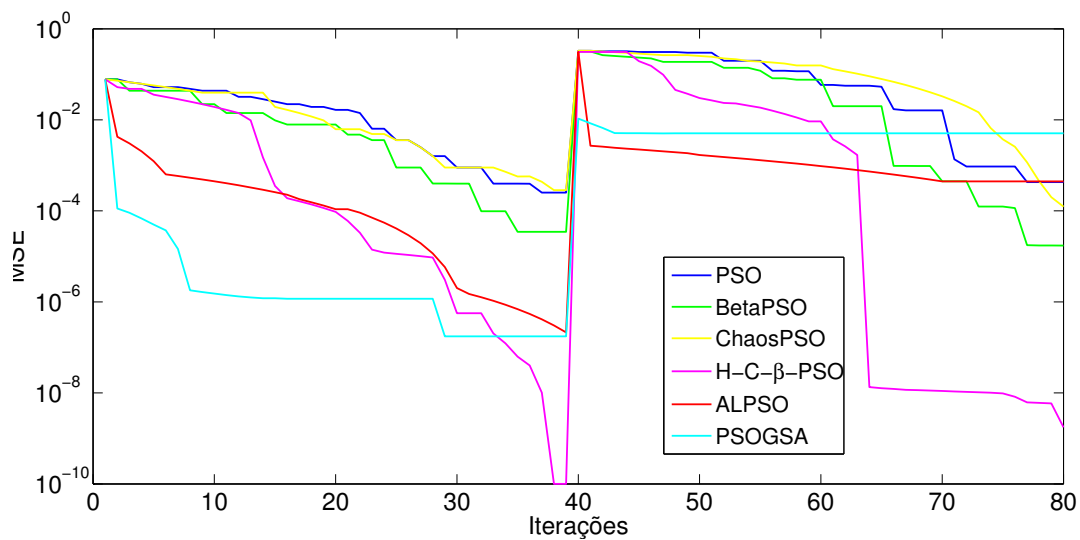


Figura 51: MSE para todos os algoritmos de estimação de parâmetros com média de 50 realizações.

Fonte: Autoria própria

Tabela 22: Número de FLOPs para o problema de estimação de parâmetros do MIT, obtidos com o programa COUNT FLOPS (QIAN, 2015).

Método	Algoritmo	FLOPs
Heurística evolucionária	PSO	6,9001e4
	β -PSO	6,8934e4
	C-PSO	7,5935e4
	PSOGSA	7,0963e4
	HC-β-PSO	6,9574e4
Híbrido Heurístico-Analítico	ALPSO	7,2109e4
Analítico Puro	ALM	1,3764e5

outros algoritmos heurísticos, uma vez que a convergência dessa abordagem necessitou de um número menor de iterações. O maior número necessário de FLOPs apresentado pelo C-PSO se deve à geração do coeficiente caótico do mapa. Por sua vez, o custo de maior complexidade do ALM vem das primeiras iterações; apesar da excelente qualidade da solução apresentada por este método, ele apresenta um custo muito elevado quando comparado a outros algoritmos; portanto, o ALM resultou em apenas uma compensação média de desempenho-complexidade. Por fim, o maior consumo de FLOPS do método ALPSO deve-se principalmente ao maior custo computacional do fitness e à necessidade de atualização dos parâmetros do método Lagrangeano aumentado.

4.5 CONCLUSÕES DO CAPÍTULO

Neste Capítulo, a metaheurística Halton-Chaos- β -PSO, foi proposta para melhorar a qualidade das soluções e a velocidade de convergência em relação ao problema de estimação de parâmetros de um motor de indução trifásico. O algoritmo proposto utilizou o método de distribuição β ao invés da distribuição uniforme do PSO tradicional e um fator de amortecimento relacionado à atualização do melhor valor global foi introduzido com sucesso. Além disso, a sequência de Halton foi usada para melhorar o mecanismo de geração inicial da população na primeira etapa para cobrir o espaço viável de forma adequada. Essas características afetaram a velocidade de convergência, o compromisso entre desempenho e complexidade e a qualidade das soluções do algoritmo. Os resultados numéricos demonstraram a eficácia do método HC- β -PSO, bem como do método analítico ALM e o método analítico-heurístico ALPSO para o problema de estimação de parâmetros do MIT. Além disso, houve uma melhoria considerável em termos do compromisso entre desempenho-complexidade para os algoritmos HC- β -PSO e ALPSO em comparação com o ALM analítico e o PSO tradicional. Apesar do bom desempenho do ALM analítico, sua complexidade computacional se demonstrou muito superior aos demais algoritmos avaliados. Por fim, o ganho proporcionado pelo o HC- β -PSO foi notável, visto que o mesmo não trouxe complexidade computacional crescente quando comparado com o PSO, β -PSO e, C-PSO na resolução do problema de estimação de parâmetros do MIT.

5 METAHEURÍSTICA HALTON-CHAOS- β -PSO NA ALOCAÇÃO DE POTÊNCIA EM REDES PON-OCDMA

Um esquema fundamental para redes móveis de banda larga é a rede ótica passiva (PON) (LIU et al., 2015). A tecnologia PON-OCDMA possui características como alta flexibilidade de rede, operação assíncrona, controle de rede simplificado, transparência de protocolo, implementação de mecanismos de qualidade de serviço (QoS) em camadas baixas, bem como aspectos aprimorados de segurança (FOULI; MAIER, 2007) (BRÈS; PRUCNAL, 2007).

As políticas de alocação de potência, visando otimizar a potência transmitida em PON-OCDMA, geralmente são empregadas para maximizar a eficiência energética, levando em consideração as restrições de QoS e considerando o SNIR de cada classe de usuário ótico. Existem várias abordagens para implementar a alocação de potência em redes PON-OCDMA, como procedimentos numéricos, inversão de matrizes, algoritmos iterativos analíticos e esquemas heurísticos (ZULAI et al., 2015). A alocação de potência em PON-OCDMA resulta em um problema de otimização não convexa, com funções objetivo e restrições não convexas. Assim, esse problema não pode ser resolvido diretamente pela implantação de técnicas convencionais de otimização convexa (DURAND; ABRÃO, 2017). A importância das metaheurísticas bioinspiradas em redes e comunicações pode ser notada a partir da extensa pesquisa de literatura, como em Kampstra et al. (2006), que apresenta uma lista de referências e respectivas aplicações de abordagens bioinspiradas para resolver problemas de telecomunicações.

Este capítulo concentra-se em comparar a capacidade, eficácia e eficiência do algoritmo Halton-Chaos- β -PSO com o PSO tradicional, versões melhoradas desse algoritmo e o método ALM-PSO, descritos nos Capítulos 4 e 5, para a resolução do problema de alocação de recursos em sistemas PON-OCDMA.

5.1 PROBLEMA DE ALOCAÇÃO DE POTÊNCIA EM SISTEMAS PON-OCDMA

Esta seção aborda o problema de alocação de recursos em redes PON-OCDMA considerado neste trabalho. A alocação ótima de potência permite melhorar a capacidade do sistema e o uso eficiente dos recursos disponíveis na rede. A alocação de potência mínima

ideal para garantir o QoS mínimo que permite a detecção correta do sinal óptico para todos os usuários (nós ópticos) é baseada na métrica da razão portadora para ruído mais interferência (CNIR).

5.1.1 ALOCAÇÃO DE TAXA DE POTÊNCIA

O QoS mínimo para um usuário específico pode ser garantido pelo SNIR medido no nó receptor i maior ou igual ao SNIR alvo mínimo exigido pela rede OCDMA e atinge um QoS aceitável por usuário associado ao BER máximo tolerável por usuário de cada classe de serviço $\gamma_i \geq \gamma_i^*$, no qual o i -ésimo SNIR é calculado por:

$$\gamma_i = \frac{r_c}{r_i} \times \Gamma_i \quad (66)$$

sendo r_i a taxa de informação alcançável (bits/s) para o i -ésimo nó, r_c é a taxa de chip do sistema disponível (largura de banda), e Γ_i é a razão entre a potência da portadora e a potência total da interferência (CIR) no i -ésimo nó, sendo calculado por (DURAND; ABRÃO, 2013):

$$\Gamma_i = \frac{G_{ii}p_i}{\sum_{j=1, j \neq i}^K G_{ij}p_j + \sigma_i^2} \quad (67)$$

sendo G_{ii} o ganho dos pares transmissor-receptor do nó óptico de interesse, p_i a respectiva potência transmitida no i -ésimo nó, p_j representa a potência transmitida pelo j -ésimo nó vizinho (interferente), σ_i^2 é a potência de ruído no i -ésimo nó receptor; os termos G_{ij} compõem os elementos dos ganhos da rede de interferência, entre o i -ésimo e o j -ésimo nós. Ademais, o efeito de emissão espontânea no pre-amplificador óptico é a principal potência de ruído à entrada do receptor (TARHUNI et al., 2006). Assim, σ_i^2 poder ser definido por:

$$\sigma_i^2 = 2 n_{sp} \cdot h \cdot f \cdot (\mathfrak{G}_i - 1) \cdot B_0 \quad (68)$$

considerando os nós de polarização no modo de fibra único, o fator de emissão espontânea é n_{sp} , f representa a frequência da operadora, h representa a constante Planck, B_0 é a largura de banda óptica e $\mathfrak{G}_i$ é o ganho do amplificador (TARHUNI et al., 2006).

A otimização da alocação de potência em sistemas OCDMA, considerando a CNIR como restrição, apresenta conexão com a taxa mínima de dados em cada nó óptico. Assim, este problema de otimização pode ser definido pela alocação de potência mínima que garante o sinal óptico adequado detectado por todos os dispositivos ópticos com requisitos de QoS atingidos (DURAND; ABRÃO, 2017). Dito de outra forma, o controle de potência em PON-OCDMA pode ser formulado como um problema de otimização de potência, sendo definido

por (TARHUNI et al., 2006):

$$\begin{aligned}
 & \underset{\mathbf{p} \in \mathbb{R}_+^K}{\text{minimize}} & J_1(\mathbf{p}) &= \mathbf{1}^T \mathbf{p} \\
 & \text{s.t.:} & \Gamma_i &= \frac{G_{ii} \cdot p_i}{\sum_{j=1, j \neq i}^K G_{ij} \cdot p_j + \sigma_i^2} > \Gamma_i^* \\
 & & p_{\min} &\leq p_i \leq p_{\max} \quad \forall i = 1, \dots, K,
 \end{aligned} \tag{70}$$

sendo $\mathbf{1}^T = [1, \dots, 1]$ o vetor unitário e Γ_i^* a razão CIR mínima que garante a QoS desejada; $p_{\max}$ e $p_{\min}$ representam os valores máximo e mínimo adotados como permitidos para a transmissão de potência, respectivamente. Utilizando a notação matricial podemos re-escrever (70):

$$[\mathbf{I} - \mathbf{\Gamma} \cdot \mathbf{H}] \geq \mathbf{u},$$

no qual $\mathbf{I}$ é a matriz identidade, $\mathbf{H}$ corresponde a matriz normalizada de interferência, com elementos avaliados de acordo com $H_{ij} = \frac{G_{ij}}{G_{ii}}$ para $i \neq j$ e zero caso contrário. Assim, $u_i = \Gamma_i \cdot \frac{\sigma_i^2}{G_{ii}}$, no qual existe uma versão escalar da potência de ruído. Realizando a substituição de desigualdade por igualdade em (70), a solução para o *vetor de potência ótimo* pode ser obtida analiticamente através da inversão da matriz (ZULAI et al., 2015):

$$\mathbf{p}^* = [\mathbf{I} - \mathbf{\Gamma} \cdot \mathbf{H}]^{-1} \mathbf{u} \tag{71}$$

No entanto, a inversão da matriz resulta em um procedimento custoso quando o número de nós cresce, resultando em um compromisso desempenho-complexidade desfavorável em sistemas ópticos CDMA de interesse prático (DURAND; ABRÃO, 2013) (TARHUNI et al., 2006).

Neste contexto, formulamos o problema de alocação de potência para K canais ópticos em sistemas OCDMA com dois objetivos enquanto o QoS do sistema, avaliado pela condição $\gamma_i \geq \gamma_i^* \forall i$ deve ser garantido:

O.I) minimizar a soma de potência da potência óptica transmitida $\sum_{i=1}^K p_i$;

O.II) atingir todos os grupos distintos de taxas de informação exigidas no sistema maiores ou iguais aos respectivos valores mínimos exigidos, $R^l \geq R_{\min}^l$.

Uma formulação alternativa para o problema de otimização é discutida em (DURAND; ABRÃO, 2013) (DURAND; ABRÃO, 2017), e adotada neste trabalho com modificações, a fim de incluir a potência óptica e taxa de dados alocados por classes de usuário:

$$\begin{aligned}
& \underset{\mathbf{p} \in \mathbb{R}_+^K}{\text{maximize}} & J_2(\mathbf{p}) &= \frac{1}{K} \sum_{l=1}^L \sum_{k=1}^{K_l} \mathcal{F}_{k,l}^{\text{th}} \cdot \left(1 - \frac{p_k^l}{p_{\max}^l}\right) \\
& \text{s.t.:} & \gamma_k &\geq \gamma_l^*, \\
& & R^l &= R_{\min}^l, \\
& & 0 < p_k^l &\leq p_{\max}^l, \quad k \in K_l, \quad l = 1, 2, \dots, L
\end{aligned} \tag{73}$$

sendo L o número de grupos distintos de taxas de informação permitidas no sistema; K_l o número de usuários para o l -ésimo grupo de taxas que tem uma taxa mínima definida por $R_{\min}^l$; finalmente, a função de limiar é dada por:

$$\mathcal{F}_{k,l}^{\text{th}} = \begin{cases} 1, & \gamma_k^l \geq \gamma_l^* \\ 0, & \text{otherwise} \end{cases}, \quad l = 1, 2, \dots, L \tag{74}$$

5.2 RESULTADOS NUMÉRICOS

Os resultados numéricos obtidos via simulação computacional são analisados nesta seção, considerando valores de parâmetros típicos para os dispositivos de rede ótica e fibra padrão, conforme apresentados na Tabela 23. No cenário considerado, o comprimento do enlace entre os nós (Tx) e o acoplador estrela e o comprimento do enlace entre os nós (Rx) e o acoplador estrela, considerando que os 16 nós estão uniformemente distribuídos em uma área com um raio entre 2 e 50 km; portanto, o intervalo do comprimento total do link é [4; 100] km. Esses comprimentos de link cobrem distâncias PON tradicionais e configurações PON de longo alcance.

Tabela 23: Valores dos Parâmetros da Rede PON-OCDMA Network

Parâmetro	Valor	Unidade
Frequência	$f = 193,1$	[THz]
Largura de Banda	$B_0 = 100$	[GHz]
Ganho	$G = 20$	[dB]
n_{SP}	2	
Constant de Planck	$h = 6,63 \times 10^{-34}$	[J/Hz]
MLEWHPC code	comprimento triplo	
Chip period	$T_C = 9$	[ps]
Potência máxima do laser	$p_{\max} = 10$	[dBm]
Potência mínima do laser	$p_{\min} = p_{\max} - 90$	[dBm]
Comprimento do link	faixa [4; 100]	[km]
# nós (usuários)	$\mathcal{K} \in \{8; 16; 32; 64\}$	(unif. distr.)
SNIR alvo	20	[dB]

O código MLEWHPC de comprimento triplo é caracterizado pelos seguintes parâmetros (KWONG; YANG, 2005; SANTOS et al., 2018):

$$(9 \times \{11, 121, 1331\}, \{1, 1, 1\}, \{1, 1, 1, 1\}, \{96/4716, 264/4716, 4356/4716\}),$$

com $k = 3, p_1 = p_2 = p_3 = 11, w = p' = 3, t_2 = 1$ e $t_1 = 3$. Além disso, para $T_C = 9$, são consideradas três classes de usuários, mostrados na Tabela 24.

Tabela 24: Classes de tráfego e parâmetros de código

Classes	Comprimento	ϕ	Duração do Bit	Taxa de Bit	BER
Classe 1	1331	4356	$p_1 p_2 p_3 T_C$	100 Mbps	BER_1
Classe 2	121	264	$p_2 p_3 T_C$	1 Gbps	BER_2
Clases 3	11	96	$p_2 p_3 T_C$	10 Gbps	BER_3

O tráfego na Classe 1 precisa de menor prioridade e taxa de bits, como voz digitalizada; A Classe 2 requer prioridade e taxa de bits moderadas, pois trata-se de tráfego de dados, enquanto a Classe 3 exige prioridade máxima e da taxa de bits mais elevada, pois comporta serviços como comunicações de vídeo de alta definição. De forma complementar, a distribuição de BER adotada é dada por $BER_3 < BER_2 < BER_1$ com pelo menos uma ordem de magnitude entre duas classes consecutivas. Sem perda de generalidade, neste trabalho foi considerada a combinação de 4 usuários da Classe 1, 4 usuários da Classe 2 e 8 usuários da Classe 3. O valor das variáveis de consumo de energia utilizadas para realizar o cálculo do consumo de energia para cada caminho de código óptico estão resumidos na Tabela 25.

5.2.1 CALIBRAÇÃO DOS PARÂMETROS DE ENTRADA DA METAHEURÍSTICA PSO

Os parâmetros utilizados pelo algoritmo PSO em todas as simulações estão descritos na Tabela 26. A definição desses parâmetros foi realizada em (DURAND; ABRÃO, 2013) para redes CDMA óticas; já o desempenho do PSO é bastante dependente da escolha dos parâmetros de entrada. Além disso, os algoritmos β -PSO, C-PSO e o Halton-Chaos- β -PSO resultam em qualidade semelhante de soluções em termos de compromisso entre desempenho-complexidade para os mesmos parâmetros de entrada. Portanto, esses parâmetros são utilizados para todos os algoritmos de otimização de alocação de taxa de potência, sendo K igual ao número de usuários. Os resultados numéricos para o ajuste dos parâmetros de entrada dos algoritmos são discutidos a seguir.

Tabela 25: Variáveis de consumo de potência

Parâmetro	Valor Adotado
η_{EO} - Eficiência de conversão de energia	1%
η_T -Eficiência do transponder	5 (W/Gbps)
E_{op}^G - Energia por operação de porta	$0,69p_tV^2$ (fJ)
E_{op}^R - Energia por registro de acesso	$3,43p_tV^2$ (fJ)
c_g - Proporção média de 1s por coluna	0,5
E_{DEC} - Consumo de energia para decodificação	8,6 (pJ/bit)
p_t - largura do canal CMOS	100 (nm)
V - Tensão de alimentação CMOS	0,8 (V)
V_{cc} - Tensão de alimentação do driver do modulador	5,2 (V)
V_{pp} - Tensão de pico de oscilação do modulador	8 (V)
R_T - Resistência de terminação do modulador	50(Ω)
L_{mod} - Perda de excesso ótico	2,5 (dB)
$\mathcal{R}$ - Responsividade do fotodiodo	0,9 (A/W)
V_{bias} - Tensão de polarização do fotodiodo	3,3 (V)
η_{EPCE} - eficiência de conversão EDFA	1%
P_{CP} - Consumo do Plano de controle	300 (W)
P_{CN} - Consumo do nó processador	17 (W)

Tabela 26: Valores dos Parâmetros do PSO

Parâmetros	Valores
Número de partículas ($\mathcal{S}$)	$K + 20$
Aceleração da partícula (c_1)	1,8
Aceleração global (c_2)	2,0
Peso inicial da inércia (ω_i)	1,0
Peso final da inércia (ω_f)	0,4
Peso da inércia (ω)	$(\omega_i - \omega_f) \left(\frac{\mathcal{I} - t}{\mathcal{I}} \right)^m + \omega_f$
Máximo de iterações ($\mathcal{I}_{\max}$)	$\mathcal{I} = 2000$
Velocidade máxima ($v_{\max}$)	$0,05(p_{\max} - p_{\min})$
Velocidade mínima ($v_{\min}$)	$-V_{\max}$

5.2.2 CALIBRAÇÃO DOS PARÂMETROS DE ENTRADA – METAHEURÍSTICA β -PSO

Inicialmente, a Figura 52 apresenta o número de iterações necessárias para que o algoritmo β -PSO resolva o problema da taxa de potência ótima, considerando diferentes valores de (p, q) . Esse aspecto é investigado considerando o objetivo de melhoria da velocidade de convergência do algoritmo. A Figura 52(a) apresenta o número de iterações para a convergência do algoritmo considerando diferentes valores do parâmetro p da função Beta. Após vários testes, o valor do parâmetro p foi fixado em $p = 1, 3$. Na verdade, muitas combinações de parâmetros foram avaliadas; no entanto, para fins práticos, apenas o valor mais representativo foi adotado a partir desta seção.

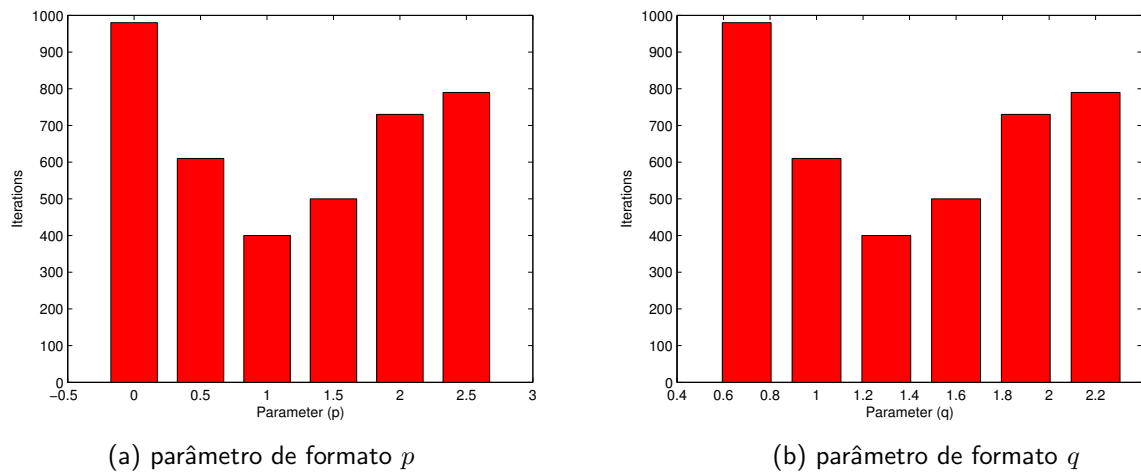


Figura 52: Número de iterações necessárias para a convergência do algoritmo β -PSO considerando diferentes valores para o parâmetro: (a) p ; (b) q

Fonte: Autoria própria

Elaborando ainda mais, o impacto do valor do parâmetro p adotado é ilustrado na Figura 52(a). Observa-se que a variação no número de iterações necessárias para a convergência do algoritmo oscilou entre ≈ 400 ($p = 1, 0$) e ≈ 1000 ($p = 0, 1$), no intervalo de avaliação adotado $p \in [0, 1; 2, 5]$. Da mesma forma, a Figura 52(b) mostra o número de iterações necessárias para a convergência do algoritmo β -PSO quando um valor diferente do parâmetro q é adotado. Nessa avaliação, o valor de p foi fixado em $p = 1, 0$ para ilustrar a metodologia de se encontrar os melhores pares (p, q) . Assim, a Figura 52(b) ilustra o comportamento do parâmetro q . Foi notado que o número de iterações para a convergência foi ≈ 400 ($q = 1, 3$), e ≈ 1400 ($q = 1, 3$), no intervalo de $p \in [0, 7; 2, 2]$. Assim, os valores $p = 1, 0$ e $q = 1, 3$ representaram o melhor compromisso entre a diversificação e intensificação para o problema PON-OCDMA usando o algoritmo β -PSO. Portanto, tais valores foram adotados no restante

da seção de resultados numéricos.

5.2.3 CALIBRAÇÃO DOS PARÂMETROS DE ENTRADA – CHAOS-PSO

Os métodos convencionais do PSO e C-PSO têm muitos parâmetros equivalentes; assim, os parâmetros do algoritmo PSO foram utilizados para o algoritmo C-PSO. Entretanto, o fator de amortecimento foi gerado usando o mapa logístico com o parâmetro de controle $\xi = 4$ para obter um comportamento caótico, como utilizado em Santana et al. (2018).

5.2.4 CALIBRAÇÃO DOS PARÂMETROS DE ENTRADA – HALTON-PSO

A Figura 53 apresenta o número de iterações necessárias para o Halton-PSO resolver o problema de alocação ótima da taxa de potência, quando diferentes valores de b são utilizados. Esse aspecto é investigado considerando o objetivo de melhoria da velocidade de convergência do algoritmo. Observa-se que todas as bases e permutações foram avaliadas para 32 usuários; no entanto, para fins práticos, apenas os valores mais representativos dos números primos foram adotados a partir desta seção. Observa-se que a variação no número de iterações necessárias para a convergência do algoritmo H-PSO oscilou entre $\mathcal{I} \approx 300$ ($b = 7$) e $\mathcal{I} \approx 650$ ($b = 23$). Esse comportamento está relacionado à variação das permutações obtidas pela alteração do número primo como base. Aqui, o valor $b = 7$ representou o melhor compromisso entre diversificação e intensificação para o problema de alocação de taxa de potência PON-OCDMA. Portanto, foi adotado $b = 7$ como base do número primo.

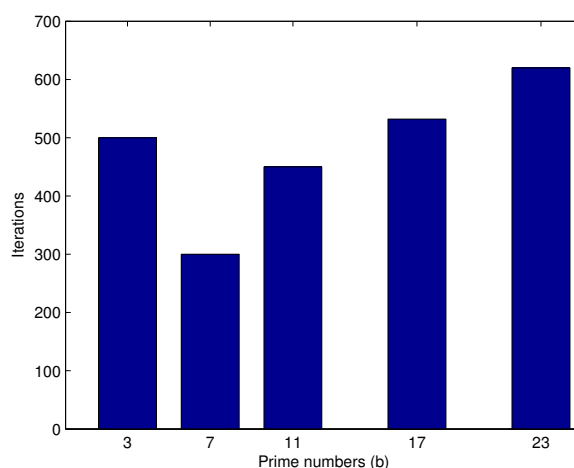


Figura 53: Número de iterações necessárias para a convergência do algoritmo H-PSO considerando diferentes valores da base b .

Fonte: Autoria própria

5.3 COMPARAÇÃO ENTRE OS MÉTODOS DE ALOCAÇÃO DE TAXA DE POTÊNCIA

Nesta seção, os algoritmos de alocação de taxa de potência em PON-OCDMA foram avaliados. O algoritmo Halton-Chaos- β -PSO foi comparado a quatro meta-heurísticas: a) PSO tradicional e suas duas variações; b) β -PSO, e c) C-PSO, bem como d) método híbrido analítico-heurístico ALMPSO. Além disso, como referência, a alocação da taxa de potência foi avaliada via inversão de matrizes, que é um método com um custo de alta complexidade, bem como o método PA ingênuo sem controle de potência, que é um método que encontra soluções marginais com um custo muito baixo custo de complexidade. Os parâmetros de rede estão descritos nas Tabelas 24, 25 e 26. A seguir, são apresentadas as análises dos algoritmos em termos de desempenho (somadas de potências, taxas da soma, qualidade de convergência), complexidade computacional e compromisso entre complexidade de desempenho.

5.3.1 SIMULAÇÕES E RESULTADOS

As simulações foram realizadas no MATLAB (versão 8.2), utilizando um computador pessoal com 8GB de RAM e processador Intel Core i5 @ 1,6GHz. O desempenho dos algoritmos foi comparado com o algoritmo PSO tradicional, validado e comparado anteriormente em (DURAND; ABRÃO, 2016, 2013; SANTANA et al., 2018). Considerando os parâmetros descritos anteriormente na Tabela 25 com três classes de serviço e $\mathcal{K} = 8, 16, 32, 64$ usuários, o desempenho dos quatro algoritmos meta-heurísticos foi comparado em termos de eficiência energética do sistema, número de iterações e FLOPs. Para realizar os testes, considerou-se os seguintes tamanhos iniciais de população: para 8 usuários $\mathcal{P} = 28$, 16 usuários $\mathcal{P} = 36$, para 32 usuários $\mathcal{P} = 52$ e para 64 usuários $\mathcal{P} = 84$.

A Figura 54 ilustra a evolução da soma de potência vs. o número de iterações obtidas pelos algoritmos PSO, β -PSO, C-PSO e Halton-Chaos- β -PSO para $\mathcal{K} = [8, 16, 32, 64]$ nós óticos distribuídos em três classes de serviço. Além disso, a figura representa a soma de potência para o caso sem controle de energia (TARHUNI et al., 2006). A linha tracejada horizontal representa a solução da soma das potências ótimas obtida com o procedimento de inversão da matrizes, o qual é efetivo na obtenção da solução, mas possui alta complexidade computacional quando comparada às soluções metaheurísticas (DURAND; ABRÃO, 2017).

Neste Capítulo, os resultados obtidos com a inversão da matrizes são usados para validar os quatro métodos de alocação da taxa de potência. Como se pode notar, para todos os cenários, os métodos convergiram para os mesmos valores de soma de potência consumida, distribuídos de forma a resultar em máxima eficiência energética (EE). Além disso, observou-

se que a evolução da soma das potências segue um comportamento monotônico suave e convergente, sendo mais acentuado (maior velocidade de convergência) nos métodos H-C- β -PSO e o híbrido ALPSO. De fato, os algoritmos HC- β -PSO e ALPSO convergiram para $\mathcal{K} = [8, 16, 32, 64]$ usuários; entretanto, a dificuldade de convergência apareceu em algumas metaheurísticas quando o número de nós aumentou. Por outro lado, quando o número de nós foi reduzido, $\mathcal{K} = 8$ e 16, todos os métodos obtiveram convergência em até 1000 iterações, com pouca variação no desempenho. Além disso, quando o número de nós óticos aumentou $\mathcal{K} = 32$ e 64, situação ilustrada nas Figuras 54(c) e 54(d), a convergência dos algoritmos H-C- β -PSO e ALPSO permaneceu mais rápida que os outros métodos, demonstrando alta capacidade de escapar dos ótimos locais, mesmo em maiores dimensões do espaço de busca.

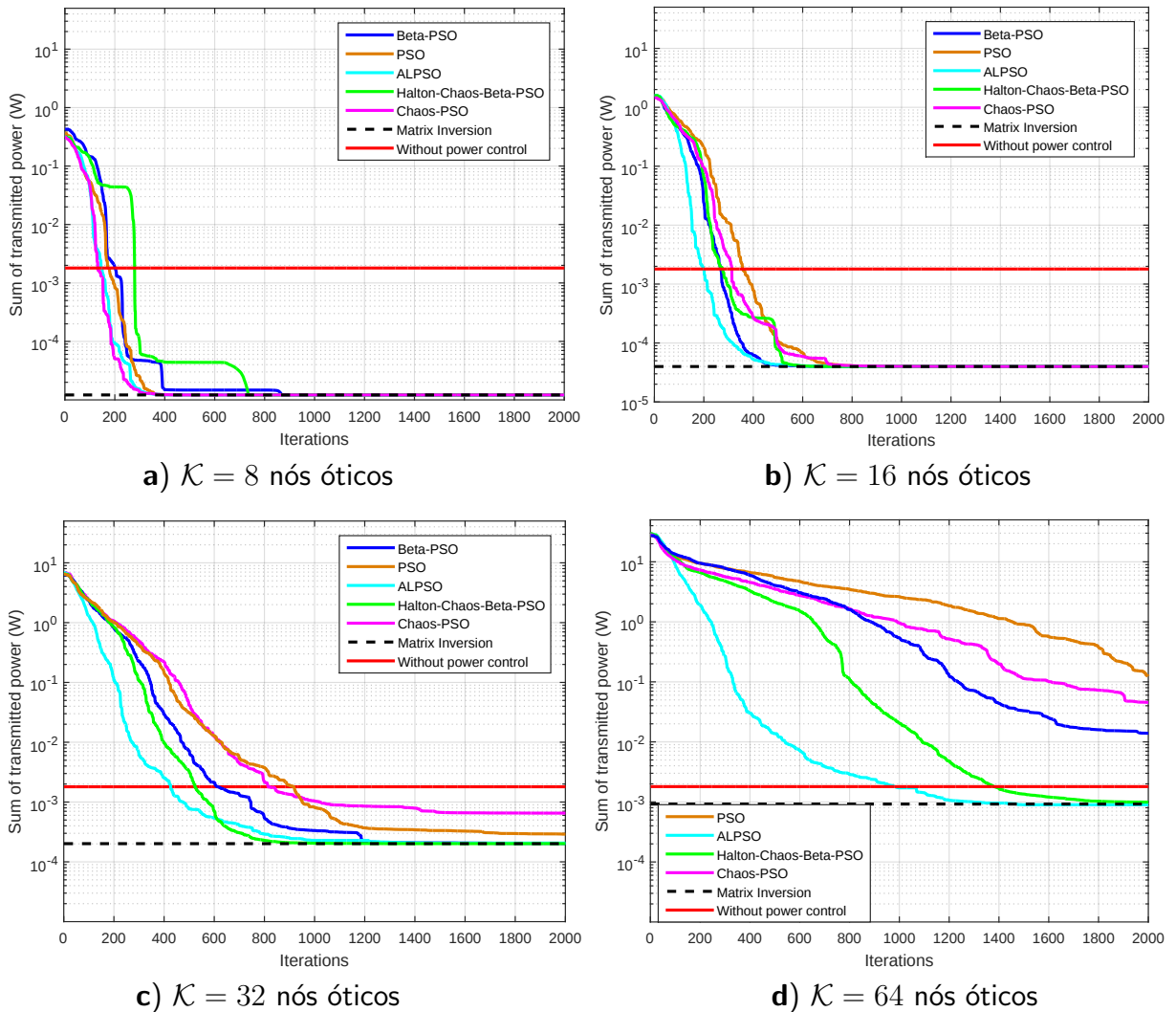
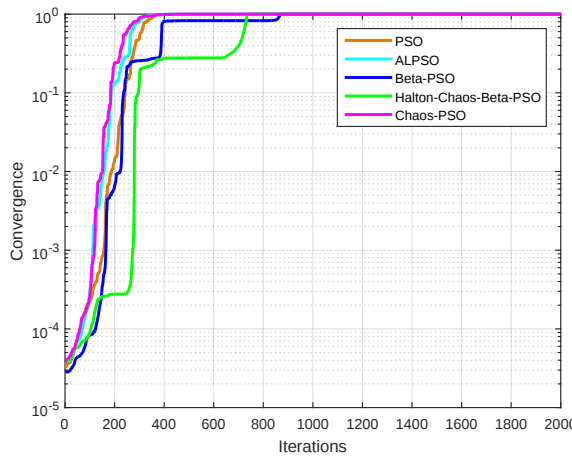


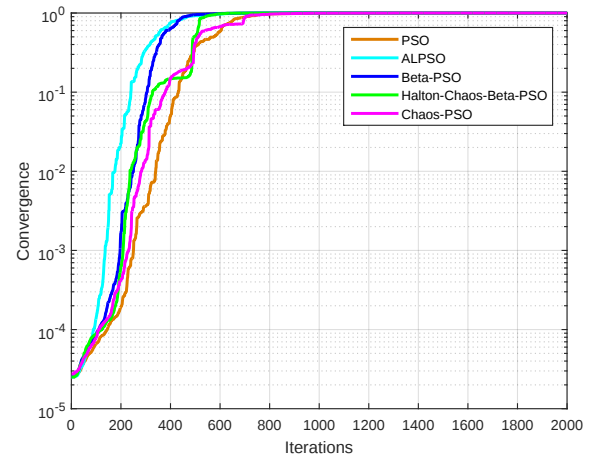
Figura 54: Soma de potência transmitida vs. número de iterações para o PSO, β -PSO, Chaos-PSO and Halton-Chaos- β -PSO, para diferentes números de nós óticos (usuários).

Fonte: Autoria própria

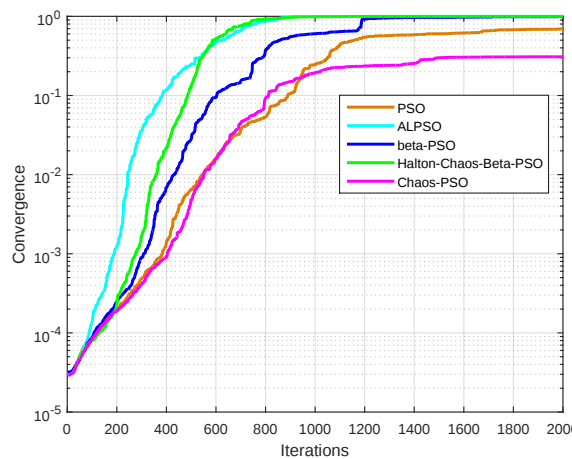
A Figura 55 mostra a convergência obtida com os cinco métodos de otimização metaheurística para o problema de alocação da taxa de potência com ($\mathcal{K} \in 8, 16, 32, 64$) nós óticos. Novamente, a melhoria da velocidade de convergência do algoritmo HC- β -PSO e o método ALPSO foi bastante superior quando comparado com o PSO tradicional, especialmente em cenários com alta dimensionalidade ($\mathcal{K} > 32$). Por exemplo, a Figura 55(c) mostra o comportamento de convergência para $\mathcal{K} = 32$ usuários, onde o número de iterações necessárias para alcançar a convergência foi $\mathcal{I} \approx 900$ para Halton-Chaos- β -PSO, enquanto para β -PSO foi $\mathcal{I} \approx 1200$. O PSO tradicional e o Chaos-PSO não alcançaram convergência para um grande número de nós óticos. Esses resultados corroboram a superioridade da metaheurística proposta, visto que a complexidade computacional associada é apenas marginalmente aumentada, como discutido na próxima seção.



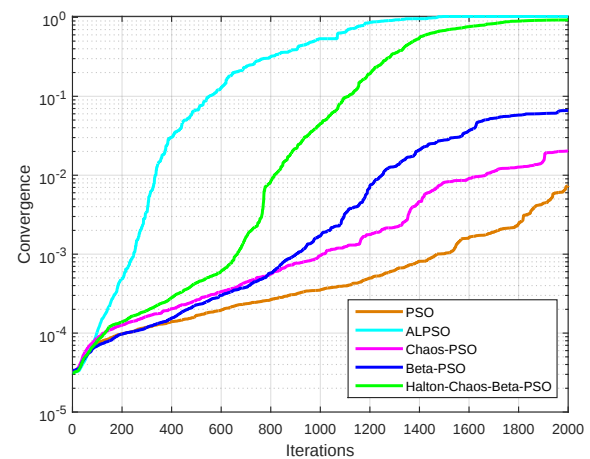
a) $\mathcal{K} = 8$ nós óticos



b) $\mathcal{K} = 16$ nós óticos



c) $\mathcal{K} = 32$ nós óticos



d) $\mathcal{K} = 64$ nós óticos

Figura 55: Convergência versus o número de iterações para os algoritmos PSO, β -PSO, Chaos-PSO and Halton-Chaos- β -PSO, para diferentes números de nós óticos (usuários).

Fonte: Autoria própria

A Figura 56 ilustra o erro quadrático médio normalizado (NMSE) versus o número de iterações dos algoritmos PSO, β -PSO, Chaos-PSO e Halton-Chaos- β -PSO para os usuários $\mathcal{K} = 8, 16, 32$ e 64 . Como pode ser observado, conforme o número de iterações aumentou, a qualidade das soluções (NMSE) foi afetada para todos os algoritmos. No entanto, o algoritmo Halton-Chaos- β -PSO apresentou menor NMSE quando comparado ao PSO tradicional e outros métodos baseados no PSO. O comportamento do platô-baixo observado nos gráficos da Figura 56 está relacionado com a não convexidade do problema de alocação de potência ótica em sistemas OCDMA. Assim, como esperado, a não convexidade do problema aumentou o número de ótimos locais (regiões de planalto). É perceptível que o Halton-Chaos- β -PSO foi mais eficaz para resolver a otimização da alocação da taxa de potência quando a dimensão do problema aumentou em comparação com as outras variantes do PSO. Neste contexto, o algoritmo proposto foi eficaz, sendo capaz de melhorar substancialmente a qualidade das soluções (considerando o mesmo número de iterações) quando comparado a outros esquemas baseados em heurísticas de alocação de taxas de potência.

Finalmente, as Figuras 57 e 58 exibem a evolução da taxa de soma e da individual dos usuário para $\mathcal{K} = 8, 16, 32, 64$ nós. A alocação da taxa de usuário se estabilizou de maneira semelhante à observada na soma da potência transmitida apresentada na Figura 54. Observa-se na Figura 58 a evolução individual da taxa de usuário atingida pelo algoritmo Halton-Chaos- β -PSO para $\mathcal{K} = 8, 16, 32, 64$ nós óticos, sendo que a taxa de dados dos usuários foi distribuída em três classes: 100 Mbps; 1 Gbps e 10 Gbps.

5.3.2 COMPLEXIDADE COMPUTACIONAL

Os FLOPs necessários para as operações principais estão resumidos na Tabela 27 e foram obtidos por meio da contagem do programa FLOPS (QIAN, 2015).

Tabela 27: Número de FLOPs para o problema de alocação de taxa de potência com o aumento do número de nós óticos.

Metaheurística	FLOPs			
	$\mathcal{K} = 8$	$\mathcal{K} = 16$	$\mathcal{K} = 32$	$\mathcal{K} = 64$
PSO	5,83638e5	1,17446e7	—	—
ALPSO	7,90361e5	1,02744e7	2,04371e8	4,15097e9
β-PSO	1,13344e6	7,70462e6	—	—
C-PSO	6,01374e5	1,19959e7	—	—
H-C-β-PSO	1,01442e6	8,18689e6	1,66790e8	4,49768e9

A Figura 59 descreve a evolução do tempo de processamento em termos do número de FLOPS consumidos pelos algoritmos PSO, ALPSO, β -PSO, C-PSO e H-C- β -PSO. A Figura

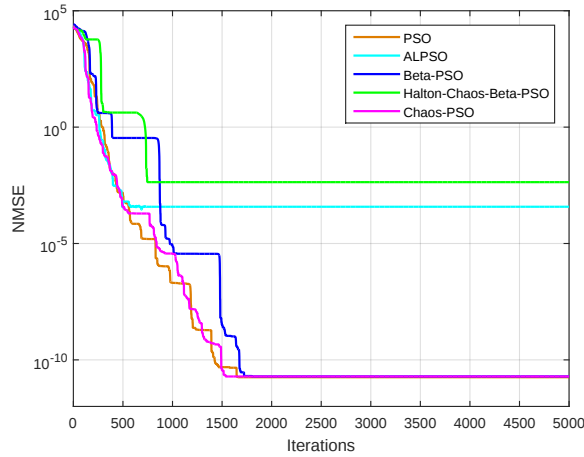
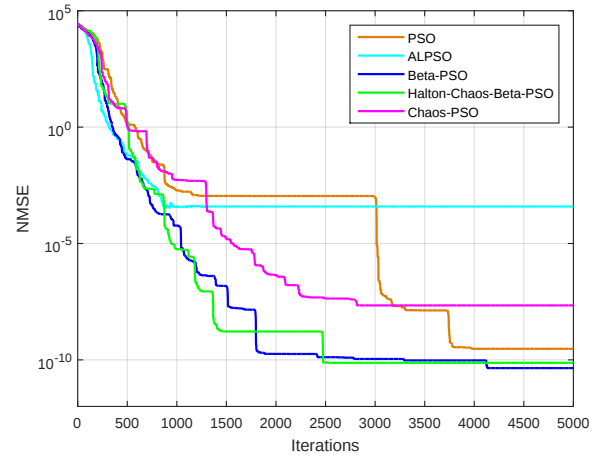
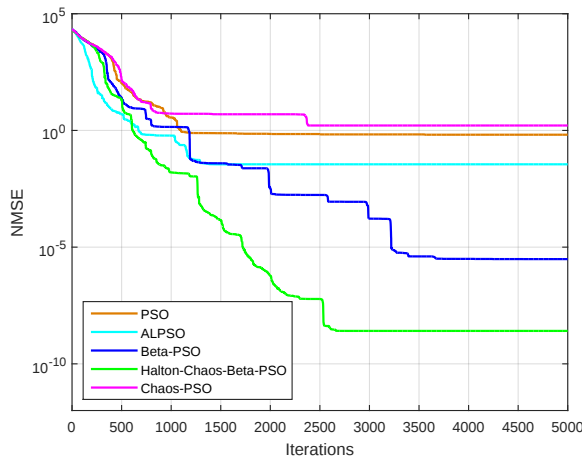
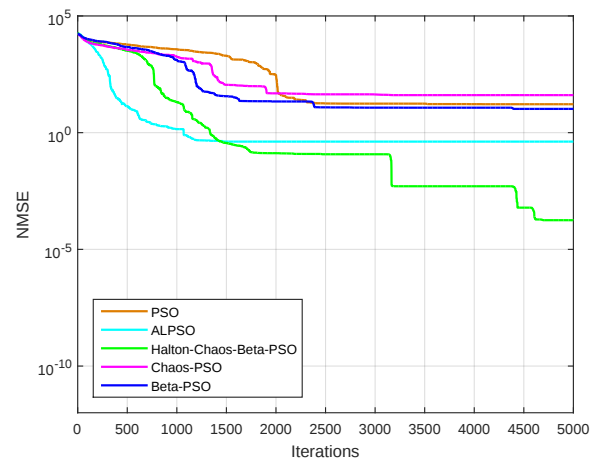
a) $\mathcal{K} = 8$ nós ópticosb) $\mathcal{K} = 16$ nós ópticosc) $\mathcal{K} = 32$ nós ópticosd) $\mathcal{K} = 64$ nós ópticos

Figura 56: NMSE versus número de iterações para os algoritmos PSO, β -PSO, Chaos-PSO and Halton-Chaos- β -PSO para diferentes números de nós ópticos (usuários).

Fonte: Autoria própria

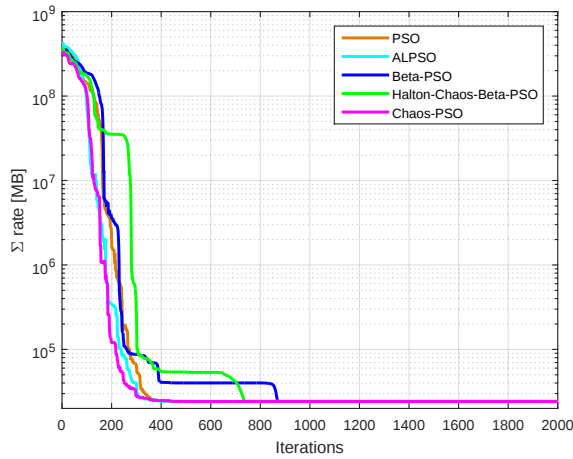
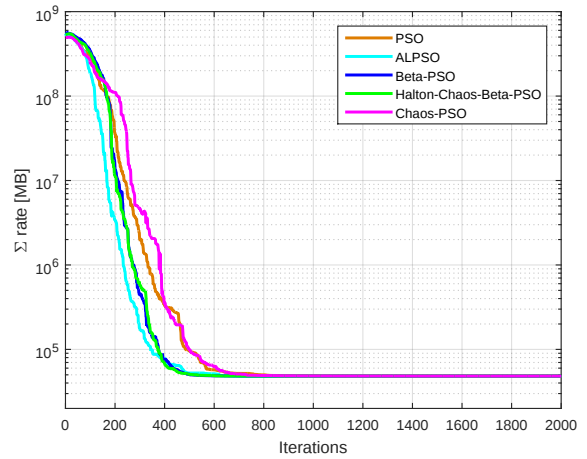
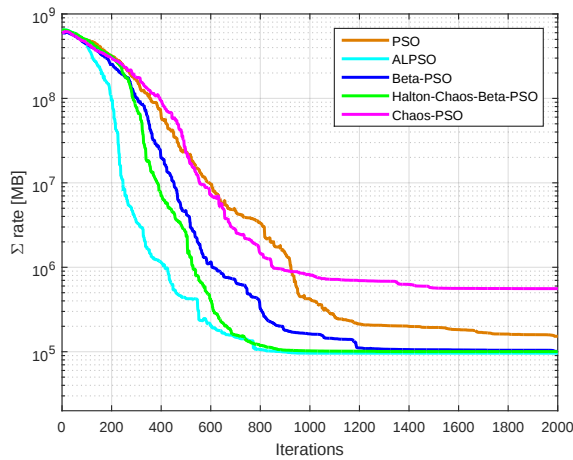
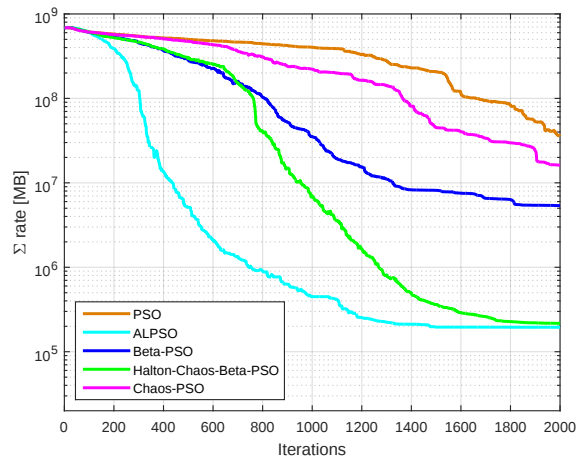
a) $\mathcal{K} = 8$ nós ópticosb) $\mathcal{K} = 16$ nós ópticosc) $\mathcal{K} = 32$ nós ópticosd) $\mathcal{K} = 64$ nós ópticos

Figura 57: Sum-rate vs the número de iterações para os algoritmos PSO, ALPSO, β -PSO, C-PSO and H-C- β -PSO para diferentes números de nós ópticos.

Fonte: Autoria própria

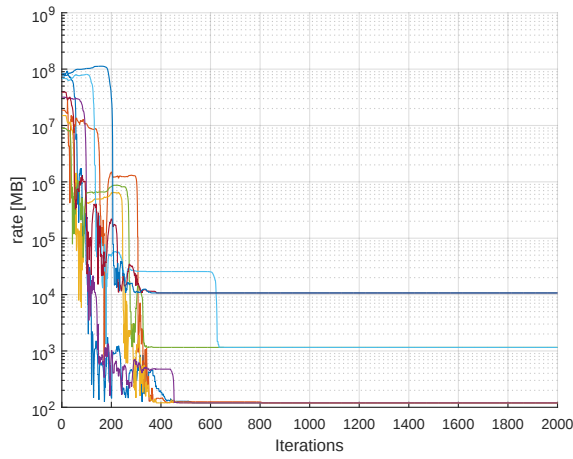
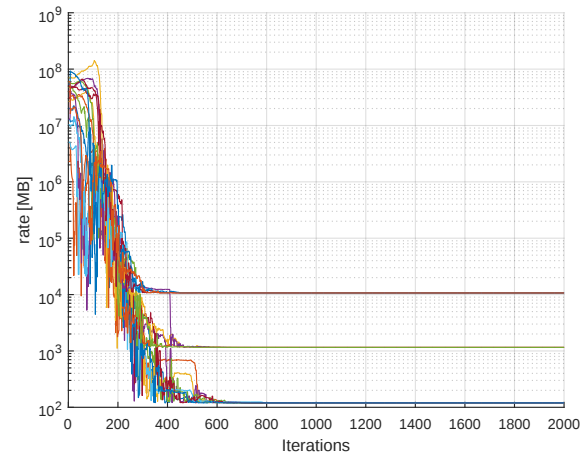
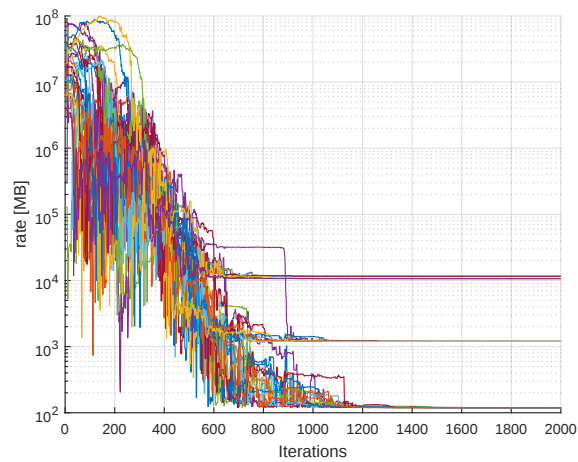
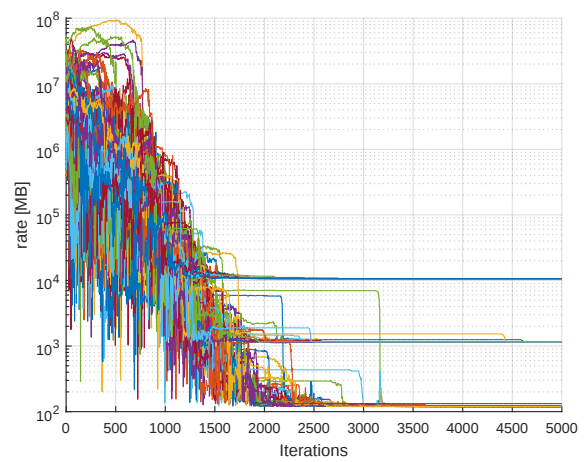
a) $\mathcal{K} = 8$ nós ópticosb) $\mathcal{K} = 16$ nós ópticosc) $\mathcal{K} = 32$ nós ópticosd) $\mathcal{K} = 64$ nós ópticos

Figura 58: Taxa individual versus número de iterações para o algoritmo Halton-Chaos- β -PSO com diferentes números de nós óptico (usuários).

Fonte: Autoria própria

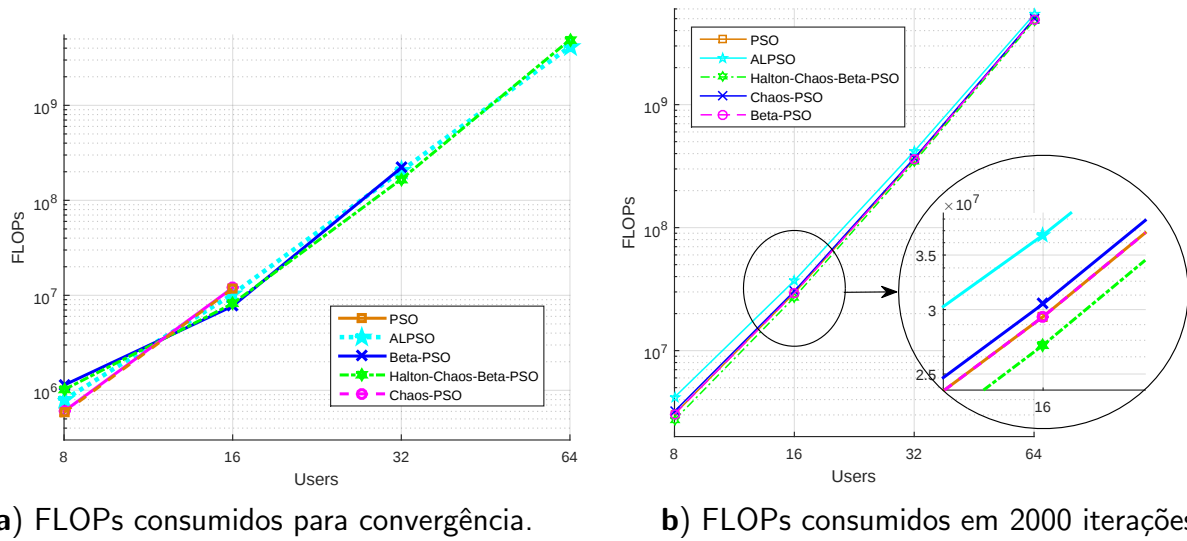


Figura 59: FLOPs consumidos com o aumento do número de nós para os algoritmos de alocação de potência.

Fonte: Autoria própria

59(a) apresenta o número de FLOPs para convergência e a Figura 59(b) mostra o número de FLOPs em 2000 iterações. O método ALPSO se mostrou como o algoritmo mais complexo, exigindo em média 15% mais FLOPs que o H-C- β -PSO, principalmente considerando a convergência prática sob um número limitado de iterações (2000). Portanto, o H-C- β -PSO foi o mais adequado para o problema de alocação de potência em redes PON-OCDMA, pois a complexidade foi menor para dimensões médias, obtendo convergência prática sob número limitado de iterações. No entanto, quando a dimensão da rede aumenta substancialmente ($\mathcal{K} > 64$), o método híbrido ALPSO torna-se mais atraente em termos de complexidade e desempenho.

Em geral, para redes pequenas e médias-altas ($8 \leq \mathcal{K} < 64$), o método H-C- β -PSO revelou um número de FLOPs em média 5% menor que o PSO e β -PSO. Para demonstrar a eficácia dos métodos HC- β -PSO, na Figura 59 considera-se a combinação de dois critérios de convergência: NMSE e número máximo de iterações. Pode-se observar que mesmo com o aumento do número de nós óticos, a complexidade em termos de FLOPs não teve um aumento considerável em relação aos demais algoritmos metaheurísticos, uma vez que a convergência do algoritmo proposto necessitou de um número menor de iterações, conforme Tabela 28. Adicionalmente, o maior número exigido de FLOPs apresentado pelo ALPSO se deve ao maior custo da função custo implantada e à necessidade de atualização periódica dos multiplicadores de Lagrange e do parâmetro de penalidade.

Tabela 28: Número de iterações necessárias para a convergência dos algoritmos considerando o aumento de nós ópticos

Método	Número de Iterações			
	$\mathcal{K} = 8$	$\mathcal{K} = 16$	$\mathcal{K} = 32$	$\mathcal{K} = 64$
PSO	382	798	—	—
ALPSO	378	554	981	1537
β -PSO	743	523	1231	—
C-PSO	374	786	—	—
H-C-β-PSO	862	601	977	1972

5.4 CONCLUSÕES DO CAPÍTULO

Neste Capítulo, a metaheurística Halton-Chaos- β -PSO, além do PSO tradicional e variações deste algoritmo, bem como o método híbrido ALM-PSO foram aplicados para resolver o problema de alocação de potência em redes PON-OCDMA, com a finalidade de melhorar a qualidade das soluções e a velocidade de convergência, i.e., o compromisso desempenho-complexidade das soluções do vetor de potência alcançados por estes cinco métodos heurístico-evolucionários.

Os resultados numéricos demonstraram a eficácia dos métodos Halton-Chaos- β -PSO e ALPSO na resolução do problema de alocação de potência ótica do sistema OCDMA, atingindo elevada eficiência energética. Além disso, obteve-se um bom compromisso entre complexidade e desempenho do algoritmo proposto em comparação ao método PSO tradicional. De fato, o incremento na velocidade de convergência foi notável, uma vez que o algoritmo Halton-Chaos- β -PSO não causou um aumento substancial na complexidade computacional em comparação aos outros algoritmos para resolução do problema de alocação de potência em redes PON-OCDMA.

6 CONCLUSÕES

Neste trabalho de Tese, foi proposta uma nova metaheurística baseada na otimização por enxame de partículas, denominada Halton-Chaos- β -PSO. A metaheurística proposta utilizou a Distribuição Beta ao invés da distribuição uniforme, utilizada pelo PSO tradicional, e um fator de amortecimento baseado em mapas caóticos para a atualização do melhor valor global encontrado no processo de otimização, aprimorando a capacidade de busca global do algoritmo ao escapar das soluções locais. Além disso, para a inicialização do enxame de partículas, foi usada a sequência de Halton ao invés da inicialização aleatória utilizada pelo algoritmo tradicional. A agregação dessas características teve como objetivo alcançar um satisfatório compromisso entre desempenho e complexidade, além de proporcionar maior velocidade de convergência e qualidade nas soluções.

Para a validação da metaheurística proposta foram utilizadas funções Benchmark, comparando seu desempenho ao obtido pelo PSO tradicional, em termos de velocidade de convergência, erro quadrático médio normalizado e número de FLOPs. Para todas as metaheurísticas comparadas foi investigada a sintonia ótima de parâmetros com o objetivo da melhoria da qualidade das soluções encontradas. Para todos os cenários e funções testadas, a metaheurística proposta apresentou o melhor compromisso entre complexidade e desempenho, alcançando um menor erro quadrático normalizado e maior velocidade de convergência.

Para avaliar a metaheurística proposta em cenários de aplicações reais, o algoritmo H-C- β -PSO foi aplicado a dois problemas de otimização de interesse prático: a) o problema de estimação de parâmetros de um motor de indução trifásico; b) problema da alocação de potência ótica mínima em sistemas OCDMA. Para o problema de estimação de parâmetros de um motor de indução trifásico, o mesmo esteve sujeito à mudança de parâmetros dessa máquina durante sua operação. Para melhor avaliação da metaheurística proposta, foram também aplicados para a resolução do problema de estimação, uma heurística analítica-iterativa que combinou o método Lagrangiano Aumentado e o ALPSO, um método ALM combinado com um método quase-Newton, e uma variante híbrida do PSO com o algoritmo de busca gravitacional. A eficiência e eficácia dos métodos híbridos analítico-iterativos-heurísticos propostos foram cor-

roboradas a partir de extensivos resultados de simulação numérica, considerando o ajuste dos parâmetros de entrada dos algoritmos metaheurísticos para cada problema tratado. Assim, observou-se uma redução notável nos valores de estimativa de erro quando este procedimento de calibração dos parâmetros de entrada das metaheurísticas foi implementado dinamicamente em aplicações em tempo real.

A metaheurística H-C- β -PSO foi aplicada com sucesso na resolução do problema de alocação de potência em redes PON-OCDMA de elevada dimensão, com a finalidade de melhorar a qualidade das soluções e a velocidade de convergência. Para o problema de alocação de potência em redes PON-OCDMA, algoritmo proposto foi comparado ao PSO tradicional, variações do PSO e ao método híbrido ALM-PSO. Os resultados numéricos demonstraram a eficácia dos métodos Halton-Chaos- β -PSO e híbrido ALPSO na resolução do problema de alocação de potência, atingindo elevada eficiência energética. Além disso, obteve-se uma melhoria considerável do compromisso desempenho-complexidade do algoritmo proposto em comparação ao método PSO tradicional e variantes disponíveis na literatura. De fato, o incremento na velocidade de convergência do algoritmo proposto foi notável, uma vez que o Halton-Chaos- β -PSO não causou um aumento substancial da complexidade computacional em comparação aos outros algoritmos para resolução do mesmo problema de alocação de potência em redes PON-OCDMA.

6.1 TRABALHOS FUTUROS

Como trabalhos futuros, sugere-se a investigação de outras formas de diversificação (randomização) das variáveis de entrada do algoritmo PSO, investigando-se outros tipos de distribuição estatística ou metodologias para a geração desses candidatos iniciais no espaço de busca. Além disso, a questão da calibração dos parâmetros de entrada pode ser realizada utilizando novas metodologias. Ademais, para a geração da população inicial do algoritmos, outras técnicas baseadas em sequências de Monte Carlo podem ser agregadas ao processo de inicialização do enxame. Também sugere-se a aplicação da abordagem proposta em outros problemas de otimização da área de Engenharia Elétrica, como: rastreamento do ponto máximo de potência em arranjos fotovoltaicos e detecção de falhas em motores de indução trifásicos.

REFERÊNCIAS

- ALATAS, B.; AKIN, E.; OZER, A. B. Chaos embedded particle swarm optimization algorithms. **Chaos, Solitons & Fractals**, Elsevier, v. 40, n. 4, p. 1715–1734, 2009.
- ALI, M. Synthesis of the β -distribution as an aid to stochastic global optimization. **Computational Statistics & Data Analysis**, Elsevier, v. 52, n. 1, p. 133–149, 2007.
- ANDRÉ, J. et al. Improving a particle swarm optimization model to solve assignment problems in an industrial environment. In: IEOM SOCIETY. **Proceedings of the 2015 international conference on operations excellence and service engineering**. [S.l.], 2015. p. 1–10.
- ARAUJO, E.; COELHO, L. d. S. Particle swarm approaches using lozi map chaotic sequences to fuzzy modelling of an experimental thermal-vacuum system. **Applied Soft Computing**, Elsevier, v. 8, n. 4, p. 1354–1364, 2008.
- AWASTHI, A. et al. An improved pso approach for optimal tuning of pi controller for shunt active power filter using fpga with hardware co-simulation. In: IEEE. **2016 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)**. [S.l.], 2016. p. 1–6.
- BALAMURUGAN, M. et al. Application of particle swarm optimization for maximum power point tracking in pv system. In: IEEE. **2016 3rd International Conference on Electrical Energy Systems (ICEES)**. [S.l.], 2016. p. 35–38.
- BANSAL, J. C. Particle swarm optimization. In: **Evolutionary and swarm intelligence algorithms**. [S.l.]: Springer, 2019. p. 11–23.
- BHOWMICK, D.; MANNA, M.; CHOWDHURY, S. K. Estimation of equivalent circuit parameters of transformer and induction motor using pso. In: IEEE. **2016 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)**. [S.l.], 2016. p. 1–6.
- BHOWMICK, D.; MANNA, M.; CHOWDHURY, S. K. Online estimation and analysis of equivalent circuit parameters of three phase induction motor using particle swarm optimization. In: IEEE. **2016 IEEE 7th Power India International Conference (PIICON)**. [S.l.], 2016. p. 1–5.
- BHOWMICK, D.; MANNA, M.; CHOWDHURY, S. K. Improved equivalent circuit parameter estimation of induction motor using hg diagram and pso. In: IEEE. **2017 IEEE Calcutta Conference (CALCON)**. [S.l.], 2017. p. 443–447.
- BHOWMICK, D.; MANNA, M.; CHOWDHURY, S. K. Estimation of equivalent circuit parameters of transformer and induction motor from load data. **IEEE Transactions on Industry Applications**, IEEE, v. 54, n. 3, p. 2784–2791, 2018.
- BLUM, C. et al. Evolutionary optimization. In: **Variants of evolutionary algorithms for real-world applications**. [S.l.]: Springer, 2012. p. 1–29.

BONYADI, M. R.; MICHALEWICZ, Z. **Particle swarm optimization for single objective continuous space problems: a review**. [S.l.]: MIT Press, 2017.

BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. **Information sciences**, Elsevier, v. 237, p. 82–117, 2013.

BRÈS, C.-S.; PRUCNAL, P. R. Code-empowered lightwave networks. **Journal of Lightwave Technology**, IEEE, v. 25, n. 10, p. 2911–2921, 2007.

CHANG, Y.-C. N-dimension golden section search: Its variants and limitations. In: IEEE. **2009 2nd International Conference on Biomedical Engineering and Informatics**. [S.l.], 2009. p. 1–6.

CHI, H.; JONES, E. L. Computational investigations of quasirandom sequences in generating test cases for specification-based tests. In: IEEE. **Proceedings of the 2006 Winter Simulation Conference**. [S.l.], 2006. p. 975–980.

CLERC, M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: IEEE. **Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)**. [S.l.], 1999. v. 3, p. 1951–1957.

COLLATZ, L.; WETTERLING, W. **Optimization problems**. [S.l.]: Springer Science & Business Media, 2012.

CUEVAS, E.; ESPEJO, E. B.; ENRÍQUEZ, A. C. Metaheuristic schemes for parameter estimation in induction motors. In: **Metaheuristics Algorithms in Power Systems**. [S.l.]: Springer, 2019. p. 9–22.

DIGEHSARA, P. A. et al. An improved particle swarm optimization based on the reinforcement of the population initialization phase by scrambled halton sequence. **Cogent Engineering**, Cogent OA, v. 7, n. 1, p. 1737383, 2020.

DORIGO, M.; CARO, G. D. Ant colony optimization: a new meta-heuristic. In: IEEE. **Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)**. [S.l.], 1999. v. 2, p. 1470–1477.

DURAND, F. R.; ABRÃO, T. Energy-efficient power allocation for wdm/ocdm networks with particle swarm optimization. **Journal of Optical Communications and Networking**, Optical Society of America, v. 5, n. 5, p. 512–523, 2013.

DURAND, F. R.; ABRÃO, T. Energy efficiency in optical cdma networks with forward error correction. **Photonic Network Communications**, Springer, v. 31, n. 1, p. 1–10, 2016.

DURAND, F. R.; ABRÃO, T. Energy efficient adaptive optical cdma random access protocol based on particle swarm optimization. **Photonic Network Communications**, Springer, v. 33, n. 3, p. 275–289, 2017.

EBERHART, R.; KENNEDY, J. Particle swarm optimization. In: CITESEER. **Proceedings of the IEEE international conference on neural networks**. [S.l.], 1995. v. 4, p. 1942–1948.

ENGELBRECHT, A. P. **Computational intelligence: an introduction**. [S.l.]: John Wiley & Sons, 2007.

FILHO, J. C. M.; SOUZA, R. N. de; ABRAO, T. Ant colony input parameters optimization for multiuser detection in ds/cdma systems. **IEEE Latin America Transactions**, IEEE, v. 12, n. 8, p. 1355–1364, 2014.

FITZGERALD, A. E. et al. **Electric machinery**. [S.l.]: McGraw-Hill New York, 2003.

FOULDS, L. R. **Optimization techniques: an introduction**. [S.l.]: Springer Science & Business Media, 2012.

FOULI, K.; MAIER, M. Ocdma and optical coding: Principles, applications, and challenges [topics in optical communications]. **IEEE Communications Magazine**, IEEE, v. 45, n. 8, p. 27–34, 2007.

GAO, W.-f.; LIU, S.-y.; HUANG, L.-l. Particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique. **Communications in Nonlinear Science and Numerical Simulation**, Elsevier, v. 17, n. 11, p. 4316–4327, 2012.

GONG, L.; CAO, W.; ZHAO, J. An improved pso algorithm for high accurate parameter identification of pv model. In: IEEE. **2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)**. [S.l.], 2017. p. 1–5.

HARRISON, K. R.; ENGELBRECHT, A. P.; OMBUKI-BERMAN, B. M. Self-adaptive particle swarm optimization: a review and analysis of convergence. **Swarm Intelligence**, Springer, v. 12, n. 3, p. 187–226, 2018.

HARRISON, K. R.; OMBUKI-BERMAN, B. M.; ENGELBRECHT, A. P. The parameter configuration landscape: A case study on particle swarm optimization. In: IEEE. **2019 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2019. p. 808–814.

HOLLAND, J. H. Genetic algorithms and adaptation. In: **Adaptive Control of Ill-Defined Systems**. [S.l.]: Springer, 1984. p. 317–333.

JONG, K. D. Evolutionary computation: a unified approach. In: **Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion**. [S.l.: s.n.], 2016. p. 185–199.

KAMPSTRA, P.; MEI, R. Van der; EIBEN, A. Evolutionary computing in telecommunication network design: A survey. **Vrije Universiteit, Faculty of Exact Sciences and CWI, Advanced Communication Networks**. Amsterdam, Netherlands, Citeseer, 2006.

KARABOGA, D.; OZTURK, C. A novel clustering approach: Artificial bee colony (abc) algorithm. **Applied soft computing**, Elsevier, v. 11, n. 1, p. 652–657, 2011.

KIRAN, M. S. Particle swarm optimization with a new update mechanism. **Applied Soft Computing**, Elsevier, v. 60, p. 670–678, 2017.

KRAUSE, P. C. Analysis of electric machinery mcgraw-hill book company. **New York**, 1986.

KRISHNAN, R. **Electric motor drives: modeling, analysis, and control**. [S.l.]: Pearson, 2001.

KWONG, W. C.; YANG, G.-C. Multiple-length extended carrier-hopping prime codes for optical cdma systems supporting multirate multimedia services. **Journal of lightwave technology**, IEEE, v. 23, n. 11, p. 3653–3662, 2005.

LEE, J. H. et al. Particle swarm optimization algorithm with intelligent particle number control for optimal design of electric machines. **IEEE Transactions on Industrial Electronics**, IEEE, v. 65, n. 2, p. 1791–1798, 2017.

LEVI, E. et al. Multiphase induction motor drives—a technology status review. **IET Electric Power Applications**, IET, v. 1, n. 4, p. 489–516, 2007.

LIU, J. et al. New perspectives on future smart fiwi networks: Scalability, reliability, and energy efficiency. **IEEE Communications surveys & tutorials**, IEEE, v. 18, n. 2, p. 1045–1072, 2015.

MACA, P.; PECH, P. The inertia weight updating strategies in particle swarm optimisation based on the beta distribution. **Mathematical Problems in Engineering**, Hindawi, v. 2015, 2015.

MARTINEZ, C. A. P. et al. Augmented lagrangian combined to evolutionary heuristic for energy efficiency in ocdma networks. **Optical Switching and Networking**, v. 36, p. 100542, 2020.

MEDJGHOU, A.; GHANAI, M.; CHAFAA, K. Improved feedback linearization control based on pso optimization of an extended kalman filter. **Optimal Control Applications and Methods**, Wiley Online Library, v. 39, n. 6, p. 1871–1886, 2018.

MIRJALILI, S.; HASHIM, S. Z. M. A new hybrid psogsa algorithm for function optimization. **2010 International Conference on Computer and Information Application**, p. 374–377, 2010.

ÑAUPARI, D. Z.; MOLINA, Y. P.; CORONEL, A. Optimal sizing of photovoltaic systems in distribution networks using particle swarm optimization. In: IEEE. **2018 IEEE PES Transmission & Distribution Conference and Exhibition-Latin America (T&D-LA)**. [S.l.], 2018. p. 1–5.

NIKCRANJBAR, A.; EBRAHIMI, M.; WOOD, A. S. Model-based fault diagnosis of induction motor eccentricity using particle swarm optimization. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, SAGE Publications Sage UK: London, England, v. 223, n. 3, p. 607–615, 2009.

ODILI, J. B. Combinatorial optimization in science and engineering. **Current Science**, JSTOR, p. 2268–2274, 2017.

ONG, C.-M. et al. **Dynamic simulation of electric machinery: using MATLAB/SIMULINK**. [S.l.]: Prentice hall PTR Upper Saddle River, NJ, 1998.

PARPINELLI, R. S. et al. A review of techniques for online control of parameters in swarm intelligence and evolutionary computation algorithms. **International Journal of Bio-Inspired Computation**, Inderscience Publishers (IEL), v. 13, n. 1, p. 1–20, 2019.

- PARSOPOULOS, K. E.; VRAHATIS, M. N. Upso: A unified particle swarm optimization scheme. In: CRC PRESS. **International Conference of Computational Methods in Sciences and Engineering 2004 (ICCMSE 2004)**. [S.l.], 2019. p. 868–873.
- PEDERSEN, M. E. H. Good parameters for particle swarm optimization. **Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001**, p. 1551–3203, 2020.
- PLUHACEK, M.; SENKERIK, R.; DAVENDRA, D. Chaos particle swarm optimization with eensemble of chaotic systems. **Swarm and Evolutionary Computation**, Elsevier, v. 25, p. 29–35, 2015.
- PRAWIN, J.; RAO, A. R. M.; LAKSHMI, K. Nonlinear parametric identification strategy combining reverse path and hybrid dynamic quantum particle swarm optimization. **Nonlinear Dynamics**, Springer, v. 84, n. 2, p. 797–815, 2016.
- PRINCE, S. K. et al. Power quality enhancement in a distribution network using pso assisted kalman filter—based shunt active power filter. In: IEEE. **2018 IEEMA Engineer Infinite Conference (eTechNxT)**. [S.l.], 2018. p. 1–6.
- QIAN, H. Counting the floating point operations (flops). **Matlab central file exchange**, v. 23, 2015.
- RAHMAN, I. U. et al. An n-state switching pso algorithm for scalable optimization. **Soft Computing**, Springer, v. 24, n. 15, p. 11297–11314, 2020.
- RASHEDI, E.; HOSSEIN, N.; SAEID, S. Gsa: A gravitational search algorithm. **Inf. Sci.**, v. 179, n. 13, p. 2232–2248, 2009.
- REN, Y.; LIU, S. Modified particle swarm optimization algorithm for engineering structural optimization problem. In: IEEE. **2017 13th International Conference on Computational Intelligence and Security (CIS)**. [S.l.], 2017. p. 504–507.
- RENNER, G.; EKÁRT, A. Genetic algorithms in computer aided design. **Computer-aided design**, Elsevier, v. 35, n. 8, p. 709–726, 2003.
- REZAEI, F.; SAFAVI, H. R. Guaspso: a new approach to hold a better exploration–exploitation balance in pso algorithm. **Soft Computing**, Springer, v. 24, n. 7, p. 4855–4875, 2020.
- SAHOO, S. K. et al. A robust accelerated pso mppt for photovoltaic system. In: **Soft Computing for Problem Solving**. [S.l.]: Springer, 2019. p. 277–288.
- SANTANA, G. A.; DURAND, F. R.; ABRÃO, T. Power allocation in pon-ocdma with improved chaos particle swarm optimization. **Journal of Microwaves, Optoelectronics and Electromagnetic Applications**, SciELO Brasil, v. 17, n. 2, p. 268–283, 2018.
- SANTOS, L. R. Rodrigues dos; DURAND, F. R.; ABRÃO, T. Adaptive power control algorithm for dynamical transmitted power optimization in mixed-line-rate optical networks. **IEEE Communications Letters**, v. 22, n. 10, p. 2032–2035, 2018.
- SEDIGHIZADEH, D. et al. Gepso: A new generalized particle swarm optimization algorithm. **Mathematics and Computers in Simulation**, Elsevier, v. 179, p. 194–212, 2021.

ŞENEL, F. A. et al. A novel hybrid pso–gwo algorithm for optimization problems. **Engineering with Computers**, Springer, v. 35, n. 4, p. 1359–1373, 2019.

SENGUPTA, S.; BASAK, S.; PETERS, R. A. Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives. **Machine Learning and Knowledge Extraction**, Multidisciplinary Digital Publishing Institute, v. 1, n. 1, p. 157–191, 2019.

SHI, Y.; EBERHART, R. A modified particle swarm optimizer. In: IEEE. **1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)**. [S.l.], 1998. p. 69–73.

SONG, B.; XU, J.; XU, L. Pso-based extended kalman filtering for speed estimation of an induction motor. In: IEEE. **2018 37th Chinese Control Conference (CCC)**. [S.l.], 2018. p. 3803–3807.

SREEKUMAR, P.; TRAD, O. A. Pso based harmonic current control in an islanded microgrid. In: IEEE. **IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society**. [S.l.], 2018. p. 5338–5342.

SRIVASTAVA, A.; DAS, D. K.; KUMAR, A. A pso based fractional order pi (fopi) controller design for a shunt active power filter for harmonic elimination. In: IEEE. **2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)**. [S.l.], 2018. p. 1–6.

STEINERBERGER, S. Dynamically defined sequences with small discrepancy. **Monatshefte für Mathematik**, Springer, v. 191, n. 3, p. 639–655, 2020.

SUGANTHAN, P. N. Particle swarm optimiser with neighbourhood operator. In: IEEE. **Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)**. [S.l.], 1999. v. 3, p. 1958–1962.

SZCZEPANIAK, P.; RUDNICKI, M. Soft-computing methods for diagnosis and design of electrical devices. In: IEEE. **1999 IEEE Africon. 5th Africon Conference in Africa (Cat. No. 99CH36342)**. [S.l.], 1999. v. 2, p. 753–758.

TALBI, E. G. A taxonomy of hybrid metaheuristics. **Journal of Heuristics**, v. 8, n. 5, p. 541–564, 2002.

TARHUNI, N. G. et al. Power control of optical cdma star networks. **Optics Communications**, Elsevier, v. 259, n. 2, p. 655–664, 2006.

TUFFIN, B. A new permutation choice in halton sequences. In: **Monte Carlo and Quasi-Monte Carlo Methods 1996**. [S.l.]: Springer, 1998. p. 427–435.

URSEM, R. K.; VADSTRUP, P. Parameter identification of induction motors using stochastic optimization algorithms. **Applied Soft Computing**, Elsevier, v. 4, n. 1, p. 49–64, 2004.

WANG, D.; TAN, D.; LIU, L. Particle swarm optimization algorithm: an overview. **Soft Computing**, Springer, v. 22, n. 2, p. 387–408, 2018.

WU, Y.; ZHU, H. Complexity comparison between two optimal-ordered sic mimo detectors based on matlab simulations. **arXiv preprint arXiv:2003.03732**, 2020.

YANG, D.; LIU, Z.; ZHOU, J. Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization. **Communications in Nonlinear Science and Numerical Simulation**, Elsevier, v. 19, n. 4, p. 1229–1246, 2014.

YANG, X.-S. **Nature-inspired metaheuristic algorithms**. [S.l.]: Luniver press, 2010.

YANG, X.-S. **Nature-inspired optimization algorithms**. [S.l.]: Elsevier, 2014.

ZAHRA, B.; SALHI, H.; MELLIT, A. Wind turbine performance enhancement by control of pitch angle using pid controller and particle swarm optimization. In: IEEE. **2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)**. [S.l.], 2017. p. 1–5.

ZHENG, Y.-x.; LIAO, Y. Parameter identification of nonlinear dynamic systems using an improved particle swarm optimization. **Optik**, Elsevier, v. 127, n. 19, p. 7865–7874, 2016.

ZULAI, L. G.; DURAND, F. R.; ABRÃO, T. Energy-efficient next-generation passive optical networks based on sleep mode and heuristic optimization. **Fiber and Integrated Optics**, Taylor & Francis, v. 34, n. 3, p. 91–111, 2015.