



DISSERTAÇÃO DE MESTRADO

**ESTUDO COMPARATIVO DE META-HEURÍSTICAS APLICADAS AO
CONTROLE PREDITIVO BASEADO EM MODELO**

EDUARDO DE MENDONÇA MESQUITA

Brasília, Agosto de 2018

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**ESTUDO COMPARATIVO DE META-HEURÍSTICAS APLICADAS AO
CONTROLE PREDITIVO BASEADO EM MODELO**

EDUARDO DE MENDONÇA MESQUITA

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA DA
UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM
SISTEMAS MECATRÔNICOS**

APROVADA POR:

Prof. Dr. Carlos Humberto Llanos Quintero, PPMEC/UnB
Orientador

Prof. Dr. Daniel Mauricio Muñoz Arboleda, Gama/UnB
Avaliador Interno

Prof. Dr. Helon Vicente Hultman Ayala, PUC/RJ
Avaliador Externo

BRASÍLIA/DF, 14 DE AGOSTO DE 2018

de Mendonça Mesquita, Eduardo

ESTUDO COMPARATIVO DE META-HEURÍSTICAS APLICADAS AO CONTROLE PREDITIVO BASEADO EM MODELO / EDUARDO DE MENDONÇA MESQUITA. – Brasil, 2018.

122 p.

Orientador: Carlos Humberto Llanos Quintero

Dissertação(Mestrado) – Universidade de Brasília – UnB

Faculdade de Tecnologia – FT

Programa de Pós-Graduação em Sistemas Mecatrônicos – PPMEC, 2018.

1. Heurísticas. 2. Algoritmos adaptativos. 3. Testes não-paramétricos. 4. Competição CEC2017. 5. Controle preditivo. I. Carlos Humberto Llanos Quintero, orientador. II. Universidade de Brasília. III. Faculdade de Tecnologia.

Agradecimentos

Para começar, quero agradecer a Deus por todas as bênçãos derramadas em minha vida ao longo deste trabalho. Com certeza não o teria terminado sem a ajuda Dele.

Agradeço a toda minha família, em especial à minha esposa Jéssica Vitória, pelo total apoio nos momentos que mais precisei, pelo carinho e compreensão demonstrados em todo o período. Aos meus pais Edna Junqueira e Francisco Pereira que sempre me incentivaram a seguir o caminho dos estudos.

Agradecimento para o meu orientador, Carlos Llanos, por me ensinar e dedicar o seu tempo à este trabalho, pela paciência e disponibilidade em ajudar quando precisei.

Ao professor Renato Coral pelo auxílio para resolver os principais problemas encontrados neste problema e por toda bagagem teórica passada a mim.

Ao professor Leandro Coelho por nos receber em Curitiba - PR e contribuir para o trabalho com muito conhecimento.

Agradeço ao Programa de Pós-Graduação em Sistemas Mecatrônicos (PPMEC) da UNB pelo suporte disponibilizado a este trabalho.

E por último, mas não menos importante, agradeço à CAPES, ao CNPQ e à FAPDF pelo fomento, incentivo e auxílio financeiro fornecidos à este trabalho.

EDUARDO DE MENDONÇA MESQUITA

RESUMO

ESTUDO COMPARATIVO DE META-HEURÍSTICAS APLICADAS AO CONTROLE PREDITIVO BASEADO EM MODELO.

Autor: Eduardo de Mendonça Mesquita

Orientador: Prof. Dr. Carlos Humberto Llanos Quintero

Programa de Pós-Graduação em Sistemas Mecatrônicos

Brasília, 14 de Agosto de 2018

A computação bioinspirada é um ramo da computação natural que busca desenvolver algoritmos de otimização inspirados no comportamento dos seres encontrados na natureza. Nesse contexto, destacam-se os algoritmos evolucionários e os baseados na inteligência coletiva. Aqueles são inspirados pela evolução biológica dos seres e estes pelo comportamento coletivo de insetos e animais. Neste trabalho, serão colocados em prova representantes dessas duas áreas. Meta-heurísticas clássicas, como PSO e DE, e também os mais recentes representantes: GWO (lobos), MFO (mariposas), SSA (salpas), WOA (baleias jubarte), DA (libélulas), além do SCA baseado no comportamento periódico das funções seno e cosseno. Os desempenhos dessas meta-heurísticas serão comparados via testes estatísticos não-paramétricos ($N \times N$ algoritmos), os quais, são métodos livres de qualquer tipo de distribuição de dados, ou seja, diferentemente dos métodos paramétricos, não dependem dos requisitos de normalidade, homogeneidade da variância e independência dos dados. Dentro desses testes não-paramétricos foram aplicados dois procedimentos *post-hoc*: Nemenyi (1963) e Shaffer (1983), os quais auxiliam na comparação em pares (1x1) do grupo de meta-heurísticas. Além das meta-heurísticas mencionadas, outras duas formam o grupo de análise, JADE e LSHADE, que utilizam técnicas auto-adaptativas dos parâmetros do DE. Os desempenhos dessas meta-heurísticas serão analisados mediante três estudos de caso: dois problemas de controle preditivo baseado em modelo, pêndulo invertido simples e atitude de satélites, e funções mono-objetivas com restrição de fronteira da competição CEC2017. Destes estudos de caso é retirada a mediana de 51 execuções de cada problema. Nos problemas de controle preditivo os algoritmos MFO e GWO tiveram bons desempenhos. Neste trabalho foram apresentadas três novas versões do GWO e uma do MFO que superaram o resultados de seus respectivos originais nos problemas de controle.

ABSTRACT

Bioinspired computing is a branch of natural computing that seeks to develop optimization inspired by the behavior of beings found in nature. In this context, the evolutionary algorithms and those based on collective intelligence stand out. The former are inspired by the biological evolution of living beings and these by the collective behavior of insects and animals. In this work, representatives of these two areas will be put to the test. Classic meta-heuristics, such as PSO and DE, and also the most recent representatives: GWO (grey wolves), MFO (moth-flame), SSA (salps), WOA (humpback whales), DA (dragonflies), as well as SCA based on the periodic behavior of sine and cosine functions. The performance of these meta-heuristics will be compared through non-parametric statistical tests ($N \times N$ algorithms), which are free methods of any type of data distribution, that is, unlike parametric ones do not depend on the requirements of normality, homoscedasticity and independence of the data. Within these non-parametric tests, two post-hoc procedures were applied: Nemenyi (1963) and Shaffer (1983), which aid in the comparison in pairs (1x1) of the metaheuristics group. In addition, two others meta-heuristics form the analysis group, JADE and LSHADE, which use self-adaptive techniques of the DE parameters. Following these techniques will be presented modifications and adaptations of GWO and MFO that present better performances than the originals. The performance of these meta-heuristics will be analyzed through three case studies: two predictive control problems, simple inverted pendulum and attitude of satellites, and mono-objective functions with bound constraints of CEC2017 competition. From these case studies the median of 51 runs of each problem is extracted. In the predictive control problems the MFO and GWO algorithms performed well. In this work were presented three new versions of GWO and one of MFO that surpassed the results of their respective originals in the control problems.

SUMÁRIO

RESUMO	i
ABSTRACT	ii
LISTA DE FIGURAS	v
LISTA DE TABELAS	vii
LISTA DE ABREVIATURAS E ACROGRAMAS	x
LISTA DE SÍMBOLOS	xi
1 Introdução	1
1.1 Objetivos da Dissertação	4
1.1.1 Objetivo Geral	4
1.1.2 Objetivos Específicos.....	4
1.2 Contribuições do Manuscrito.....	4
1.3 Apresentação do Documento	5
2 Revisão Bibliográfica	6
2.1 Algoritmos Bioinspirados	6
2.1.1 PSO - <i>Particle Swarm Optimization</i>	7
2.1.2 DE - <i>Evolução Diferencial</i>	9
2.1.3 DE adaptativo	11
2.1.4 GWO - <i>Grey Wolf Optimizer</i> (2014).....	16
2.1.5 MFO - <i>Moth-Flame Optimization</i> (2015)	20
2.1.6 DA - <i>Dragon-Fly Algorithm</i> (2016)	23
2.1.7 WOA - <i>Whale Optimization Algorithm</i> (2016).....	26
2.1.8 SCA - <i>A Sine Cosine Algorithm</i> (2016).....	29
2.1.9 SSA - <i>Salp Swarm Algorithm</i> (2017).....	32
2.2 Análise Estatística de Desempenho.....	34
2.2.1 Princípios da estatística	34
2.2.2 Teste de hipótese	38
2.2.3 Testes paramétricos e não-paramétricos	40
2.2.4 Teste de normalidade	41

2.2.5	Teste de homogeneidade de variância	45
2.2.6	Teste não-paramétrico do desempenho	46
2.2.7	Procedimentos <i>post-hoc</i>	48
2.2.8	Software KEEL	48
2.3	Controle Preditivo baseado em Modelo (MPC).....	50
2.3.1	Metodologia do MPC	51
2.3.2	Fundamentos matemáticos do MPC	52
2.4	Conclusões do capítulo	54
3	Metodologia	55
3.1	Estudos de caso	55
3.1.1	Pêndulo Invertido em um carrinho	55
3.1.2	Satélite	58
3.1.3	<i>Benchmark</i> CEC2017	62
3.2	Método de aquisição de dados	64
3.2.1	Estratégia otimizador NMPC.....	64
3.2.2	Parâmetros	65
3.2.3	Geração dos resultados	67
3.3	Método de análise de desempenho.....	68
3.4	Modificações e adaptações.....	68
3.5	Método de análise do tempo de execução	69
3.6	Conclusões do capítulo	69
4	Resultados	70
4.1	Resultados dos estudos de caso	70
4.2	Testes de normalidade.....	70
4.3	Testes de homogeneidade da variância.....	72
4.4	Análise de múltiplas meta-heurísticas.....	74
4.4.1	Análise qualitativa	76
4.5	Meta-heurísticas propostas nesta dissertação.....	81
4.5.1	GWO - 1ª versão	81
4.5.2	GWO - 2ª versão	82
4.5.3	GWO - 3ª versão	83
4.5.4	MFO adaptativo	83
4.6	Desempenho das meta-heurísticas propostas.....	84
4.6.1	Controle do pêndulo invertido simples.....	84
4.6.2	Controle de atitude de satélites.....	86
4.6.3	Validação das adaptações de GWOv1 e MFOadap	88
4.7	Análise do tempo de execução	89
5	Conclusões	92
5.1	Conclusões do Trabalho	92
5.2	Sugestões para Trabalhos Futuros.....	93

REFERÊNCIAS BIBLIOGRÁFICAS.....	94
APÊNDICES	99
A Resultado mediana 51 simulações	100
B Resultado dos procedimentos post-hoc.....	107
ANEXOS.....	112
I Funções básicas - CEC2017	113
II Benchmark - CEC2017	117

LISTA DE FIGURAS

1.1	Métodos de solução para problemas de otimização (WEISE, 2009).....	1
1.2	Taxonomia dos algoritmos bioinspirados, <i>adaptado de</i> Binitha, Sathya et al. (2012)	2
2.1	Hierarquia entre os lobos-cinzentos (MIRJALILI; MIRJALILI; LEWIS, 2014)	17
2.2	Caça dos lobos-cinzentos: A) aproximação, (B-D) cercando a presa e E) aguardando e ataque, (MURO et al., 2011 apud MIRJALILI; MIRJALILI; LEWIS, 2014)	18
2.3	(a) refinamento e (b) exploração da busca, <i>adaptado de</i> (MIRJALILI; MIRJALILI; LEWIS, 2014)	19
2.4	Orientação transversal (a) e (b) da busca, <i>adaptado de</i> (MIRJALILI, 2015)	21
2.5	Deslocamento em espiral de um agente de busca, <i>adaptado de</i> (MIRJALILI, 2015).....	22
2.6	Deslocamento estático (a) e dinâmico (b), (MIRJALILI, 2016a)	23
2.7	Padrões de deslocamento individual em um grupo de libélulas, <i>adaptado de</i> (MIRJALILI, 2016a)	24
2.8	Rede de bolhas da baleia Jubarte, (MIRJALILI; LEWIS, 2016).....	27
2.9	Variação da região de busca conforme valor de r_1 , <i>adaptado de</i> (MIRJALILI, 2016b).....	30
2.10	Balanceamento entre exploração e refinamento no SCA, <i>adaptado de</i> (MIRJALILI, 2016b).	31
2.11	Salpa (a) e uma corrente de salpas (b) (MIRJALILI et al., 2017).....	32
2.12	Inferência estatística da população	34
2.13	Histograma de um conjunto de 10.000 dados gerados pela função $randn(\cdot)$ do MatLab	35
2.14	Função distribuição acumulada ($\mu = 0$ e $\sigma = 1$)	35
2.15	Distribuição normal	36
2.16	Distribuição Qui-Quadrado χ^2	37
2.17	Distribuição F de Snedecor	38
2.18	Intervalo de confiança para $\alpha = 0.5$	39
2.19	Distribuição conhecida $FDE_0(x)$, distribuição empírica $FDE_n(x)$	42
2.20	Tela inicial	49
2.21	Tela de módulos	49
2.22	Tela dos testes estatísticos	50
2.23	Diagrama de blocos da metodologia do MPC, <i>adaptado de</i> (CAMACHO; BORDONS, 2012)	51
2.24	Mapeamento dos estados futuros dada uma sequência de ações de controle futura \tilde{u} , <i>adaptado de</i> (ALAMIR, 2013)	53
2.25	Matriz de seleção em um caso especial de $N = 6$ (ALAMIR, 2013)	53
3.1	Pêndulo invertido da Quanser	56

3.2	a) pêndulo conectado ao carrinho, b) diagrama de corpo livre do carrinho, c) diagrama de corpo livre do pêndulo, (MERCIECA; FABRI, 2011)	56
3.3	Plataforma de testes: mesa giratória de três eixos, (RODRIGUES; MURILO; SOUZA, 2017)	58
3.4	Sinais de entrada, degrau e degrau filtrado	60
3.5	Processo interno do otimizador realizado pelo algoritmo bioinspirado	65
4.1	Convergência dos valores da função custo - pêndulo	77
4.2	Estados do pêndulo	78
4.3	Ação de controle u	79
4.4	Convergência dos valores da função custo - satélite	79
4.5	Variação nos estados e comandos	80
4.6	Resultados CEC2017 por dimensão	81
4.7	Convergência das meta-heurísticas propostas - pêndulo	85
4.8	Convergência das meta-heurísticas propostas - satélite	87
4.9	Variação de M_p - pêndulo	88
4.10	Variação dos valores de M_b em diferentes instantes	89

LISTA DE TABELAS

2.1	Ocorrência de erros no teste de hipótese.....	39
2.2	Significado do <i>valor-p</i>	40
3.1	<i>Benchmark</i> CEC2017	63
3.2	Condições iniciais - sistemas NMPC	65
3.3	Parâmetros do sistema do Pêndulo	66
3.4	Parâmetros do sistema do Satélite.....	66
3.5	Parâmetros do <i>benchmark</i> CEC2017	66
3.6	Parâmetros PSO.....	67
4.1	Resultado: testes de normalidade - Pêndulo	71
4.2	Resultado: testes de normalidade - Satélite	71
4.3	Resultado: testes de normalidade - CEC2017 com dimensão 10	71
4.4	<i>Valor-p</i> dos testes de normalidade - CEC2017 com dimensão 30	72
4.5	<i>Valor-p</i> dos testes de normalidade - CEC2017 com dimensão 100.....	72
4.6	Variância das meta-heurísticas - Pêndulo.....	73
4.7	Variância das meta-heurísticas - Satélite.....	73
4.8	Variância das meta-heurísticas - CEC2017.....	73
4.9	<i>Ranking</i> das meta-heurísticas - NMPC.....	74
4.10	<i>Ranking</i> das meta-heurísticas - CEC2017	74
4.11	<i>Overshoot</i> dos ângulos de Euler	80
4.12	<i>Ranking</i> das meta-heurísticas propostas - Pêndulo	84
4.13	Valores- <i>p</i> ajustados - novas versões - Pêndulo	85
4.14	<i>Ranking</i> das meta-heurísticas propostas - Satélite.....	86
4.15	Valores- <i>p</i> ajustados - novas versões - Satélite	87
4.16	Especificações da máquina utilizada nas simulações	89
4.17	Tempo de execução das meta-heurísticas - pêndulo	90
4.18	Tempo de execução das meta-heurísticas - satélite.....	90
4.19	Tempo de execução das meta-heurísticas - CEC2017	90
A.1	Dados das simulações para o Pêndulo.....	101
A.2	Dados das simulações para o Satélite.....	102
A.3	Dados das simulações para as funções do CEC2017 - $D = 10, (\cdot 10^3)$	103
A.4	Dados das simulações para as funções do CEC2017 - $D = 30, (\cdot 10^4)$	103

A.5	Dados das simulações para as funções do CEC2017 - $D = 100, (\cdot 10^5)$	104
A.6	Dados das simulações das modificações e adaptações propostas ao GWO e MFO aplicadas ao problema do pêndulo	105
A.7	Dados das simulações das modificações e adaptações propostas ao GWO e MFO aplicadas ao problema do satélite.....	106
B.1	Valores- p ajustados - Pêndulo	107
B.2	Valores- p ajustados - Satélite	108
B.3	Valores- p ajustados - CEC2017 - $D=10$	109
B.4	Valores- p ajustados - CEC2017 - $D=30$	110
B.5	Valores- p ajustados - CEC2017 - $D=100$	111

LISTA DE ABREVIATURAS E ACROGRAMAS

APV	valor- p ajustado
DA	Dragon-Fly Algorithm
DAP	estatística do teste D'Agostinho e Pearson
DE	Evolução Diferencial
FWER	Family-Wise Error Rate
GWO	Grey Wolf Optimizer
GWOv1	primeira versão proposta do GWO
GWOv2	segunda versão proposta do GWO
GWOv3	terceira versão proposta do GWO
JADE	Adaptative Differential Evolution
KEEL	Knowledge Extraction based on Evolutionary Learning
LEIA	Laboratory of Embedded Systems and Integrated Circuits Applications
LPSR	Linear Population Size Reduction
LSHADE	SHADE com LPSR
MFO	Moth-Flame Optimization
MFOadap	versão auto-adaptativa proposta do MFO
MPC	Controle Preditivo baseado em Modelo
NFL	No-Free Lunch Theorem
NMPC	Controle Preditivo não-linear
PSO	Particle Swarm Optimization
SCA	A Sine Cosine Algorithm
SHADE	Succes-History based Adaptative DE
SSA	Salp Swarm Algorithm
SQP	Sequential Quadratic Programming
WOA	Whale Optimization Algorithm

LISTA DE SÍMBOLOS

α	nível de significância de um teste estatístico
C_h	posição das melhores aptidões do MFO
δ_u	variação da ação de controle u
CR	parâmetro de crossover do DE
D	dimensão do problema
F	matriz das aptidões dos agentes de busca
F_f	estatística do teste Iman e Davenport
F_m	fator de mutação do DE
gbest	melhor resultado do PSO
H_0	hipótese nula
H_1	hipótese alternativa
k	número de algoritmos
lb	limite inferior do espaço de busca
Max_{iter}	número máximo de iterações
M_{CR}	média do arquivo S_{CR}
M_F	média do arquivo S_F
M_b	média dos parâmetros de b
M_p	média dos parâmetros de p
n	número de problemas
N	horizonte de predição do MPC
N_u	horizonte de controle do MPC
$pbest_i$	melhor resultado da partícula i do PSO
$pNeme$	valor- p do teste Nemenyi
$pShaf$	valor- p do teste Shaffer
Q	penalização do erro da trajetória no instante de 1 a N-1
Q_{erro}	penalização do estado x do pêndulo
Q_f	penalização do erro da trajetória no último instante de tempo
R	penalização da ação de controle do pêndulo
R_j	ranking de Friedman do algoritmo j
S	número de agentes de busca
S_{CR}	arquivo de CRs com sucesso dos algoritmos adaptativos

S_F	arquivo de Fs com sucesso dos algoritmos adaptativos
S_{ini}	número inicial de agentes de busca dos algoritmos adaptativos
S_{min}	número mínimo de agentes de busca dos algoritmos adaptativos
t	iteração atual do algoritmo
τ_s	período de amostragem
θ_{ss}	valor estacionário dos estados do satélite
ub	limite superior do espaço de busca
U	vetor de cruzamento cromossômico do DE
u	ação de controle do MPC
X	matriz dos agentes de busca
X_a	variável aleatória
X_{best}	agente de busca com melhor aptidão do DE
X_p	posição do agente com melhor aptidão dos algoritmos SCA, SSA e WOA
X_{ss}	valor estacionário dos estados do pêndulo
χ_f	estatística do teste de Friedman
V	vetor de mutação do DE
V_i	velocidade da partícula i da meta-heurística PSO
V_{max}	limite máximo de V_i
w	fator de inércia do PSO
w_0	valor inicial de w
w_f	valor final de w
W	estatística do teste de Levene
W_s	estatística do teste Shapiro-Wilk

Capítulo 1

Introdução

O problema central da computação é encontrar algoritmos que solucionem, da melhor forma, as mais variadas tarefas do cotidiano. Dentre elas, têm-se como exemplo a computação gráfica, os bancos de dados, a computação paralela, o entretenimento (jogos), a inteligência artificial e a otimização. Este trabalho está inserido nesta última, a qual busca por métodos mais eficientes em termos de solução e/ou tempo do que os já desenvolvidos até o presente (FEIJÓ; SILVA; CLUA, 2009). Esses métodos podem ser divididos de duas formas, conforme Fig. 1.1.

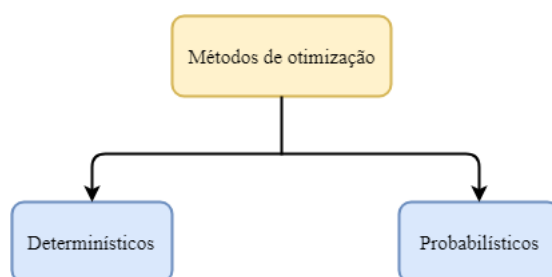


Figura 1.1: Métodos de solução para problemas de otimização (WEISE, 2009)

Os métodos determinísticos são baseados em cálculos analíticos da função de otimização, como por exemplo o *gradiente*. Eles possuem a característica de garantir a solução ótima. Entretanto, se a relação entre a solução candidata e sua função custo é complicada de ser traçada, ou a dimensionalidade do espaço de busca é muito alta, torna-se mais difícil resolver o problema de forma determinística. Assim, entra em cena os métodos probabilísticos, nos quais se encontram as meta-heurísticas. Elas são métodos de solução para classes de problemas gerais, tratando-os como procedimentos estilo *caixa preta* (WEISE, 2009). Além disso, as meta-heurísticas normalmente não realizam cálculos de gradiente e não usam nenhuma informação da derivada da função, isso auxilia no uso desses métodos para resolver problemas altamente não lineares e problemas com descontinuidade (YANG, 2017).

Um problema de otimização é classificado entre minimização ou maximização, podendo ser uma tarefa simples tal como encontrar as raízes de uma função de segundo grau, ou algo mais complexo como determinar a melhor dinâmica da asa de um avião que minimize a resistência imposta pelo ar. De qualquer forma, sempre haverá um problema mais complexo que a natureza já resolveu e muitas vezes de forma

misteriosa. Algumas tecnologias já foram criadas inspiradas na natureza: (a) o sonar é um sensor de medição inspirado no mecanismo de locomoção dos morcegos e (b) o velcro é baseado na estrutura de plantas que se prendem a determinadas superfícies através do contato.

Diante desse cenário, surge a *computação natural* como uma nova era da computação, desenvolvendo meta-heurísticas ara áreas como robótica, sistemas de controle, mineração de dados, engenharia de produção e várias outras (BINITHA; SATHYA et al., 2012). Dentro da computação natural encontra-se a *computação bioinspirada*, cujas meta-heurísticas são agrupadas pela área de inspiração, a exemplo disso na Fig. 1.2 podem ser vistas três dessas ramificações. A partir daqui, os termos *algoritmo* e meta-heurística serão usados para definir os métodos encontrados na computação bioinspirada.

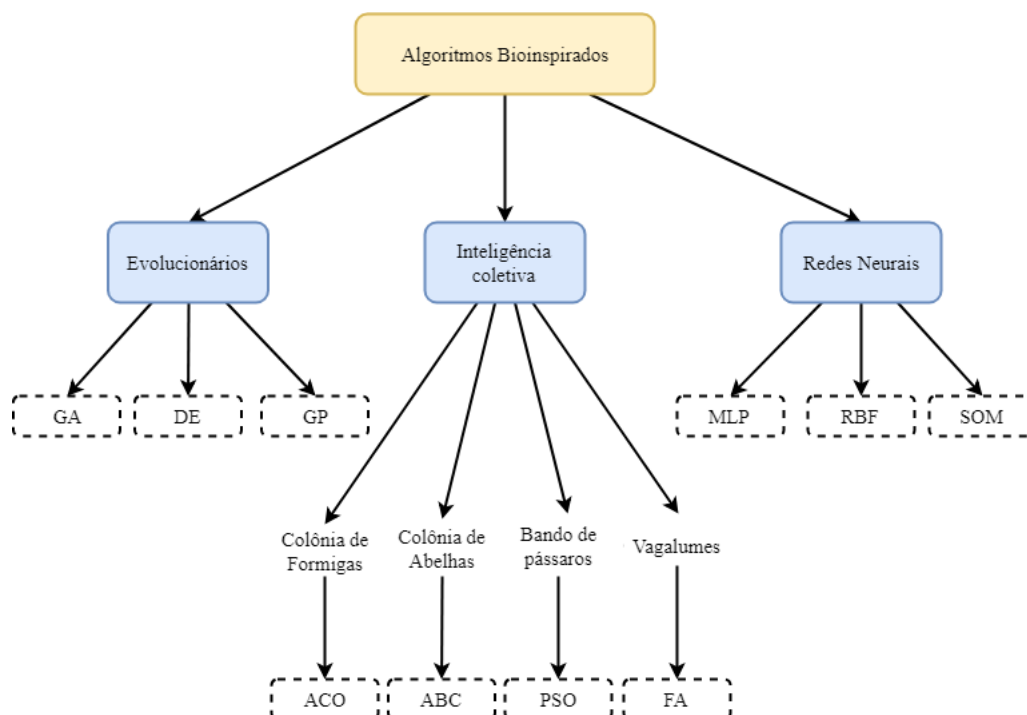


Figura 1.2: Taxonomia dos algoritmos bioinspirados, *adaptado de* Binitha, Sathya et al. (2012)

Os algoritmos evolucionários são inspirados na evolução biológica dos seres, baseados nas teorias de Darwin. Os algoritmos GA - algoritmo genético, GP - programação genética e DE - evolução diferencial são representantes de grande êxito dessa família. Outra família é a das redes neurais, as quais são algoritmos com metodologias inspiradas na estrutura neural de organismos inteligentes, geralmente pelo estudo do comportamento dos neurônios no cérebro. A exemplo dessa família de algoritmos estão: MLP - perceptron multi-camadas, RBF - rede com função de base radial e SOM - redes neurais auto-organizáveis.

Já a inteligência coletiva baseia-se no comportamento de animais e insetos quando agrupados, pois apresentam a chamada inteligência de enxame (*swarm intelligence* em inglês), designando sistemas com interação entre os indivíduos, com pouca (ou nenhuma) inteligência, mas que geram soluções coerentes e padrões de busca na realização de tarefas em conjunto (POLI; KENNEDY; BLACKWELL, 2007).

Os algoritmos evolucionários e a inteligência coletiva são as duas classes predominantes e mais bem-sucedidas dos algoritmos bioinspirados (BINITHA; SATHYA et al., 2012). Mas o que torna um algoritmo

mais bem-sucedido do que outro depende não só do método mas também do problema a ser resolvido. O teorema da inexistência do almoço grátis (NFL - do inglês *No-Free Lunch Theorem*), proposto e demonstrado por Wolpert, Macready et al. (1995), diz que todos os algoritmos de otimização aplicados em todos os problemas existentes, em média, possuem o mesmo desempenho. Em outras palavras, quer dizer que se o algoritmo A é melhor que o algoritmo B em uma porção de n problemas, então existirá outra porção de n problemas que o algoritmo B terá melhor desempenho que o algoritmo A.

A fim de avaliar o desempenho em múltiplos problemas ($n > 1$), existem técnicas de comparação entre dois algoritmos (1×1), entre um algoritmo e um grupo de algoritmos ($1 \times N$) ou entre múltiplos algoritmos entre si ($N \times N$). Dessas técnicas, existem os métodos estatísticos paramétricos e não-paramétricos. Estes são mais indicados para a análise devido à ausência de limitações quanto ao tipo de dados a serem analisados, já que os testes paramétricos pressupõem-se a normalidade, independência e homogeneidade da variância dos dados para garantir a confiabilidade dos testes (CONOVER, 1999).

O uso dos testes estatísticos não-paramétricos pode ser visto, por exemplo, na avaliação de classificadores de múltiplos dados (GARCIA; HERRERA, 2008), algoritmos evolucionários (GARCÍA et al., 2009; GARCÍA et al., 2009) e redes neurais (TRAWIŃSKI et al., 2012). Derrac et al. (2011) apresenta um tutorial do uso desses testes e mostra para cada tipo de comparação dos dados quais os testes já desenvolvidos.

Além de verificar o algoritmo que encontre as melhores soluções para determinado conjunto de problemas, o projetista/usuário muitas vezes precisa determinar a mais adequada configuração dos parâmetros. Esse tipo de procedimento não é simples e na maioria dos casos demanda um custo computacional elevado de tentativa e erro. Para resolver esse inconveniente trabalho, surgem os algoritmos adaptativos, nos quais o valor dos parâmetros variam a fim de guiar o resultado para a melhor solução.

As técnicas de adaptação dos parâmetros podem ser divididas em três categorias: (a) determinística, (b) adaptativa e (c) controle auto-adaptativo (SMITH; FOGARTY, 1997 apud QIN; HUANG; SUGANTHAN, 2009). A determinística modifica o valor dos parâmetros através de uma regra pré-definida, sem levar em conta nenhuma informação de convergência do algoritmo. A técnica adaptativa incorpora alguma informação do processo de busca para guiar a atualização dos parâmetros. Já o controle auto-adaptativo direciona os parâmetros de acordo com a aptidão do algoritmo naquele instante, ou seja, os parâmetros dos indivíduos com melhores soluções irão ser utilizados como base para a adaptação dos outros.

Por conta da necessidade de adaptação dos seus parâmetros, o DE (Evolução Diferencial) apresenta diversas versões adaptativas. O SaDE (QIN; HUANG; SUGANTHAN, 2009), JADE (ZHANG; SANDERSON, 2009) e o SHADE (TANABE; FUKUNAGA, 2013) são exemplos de auto-adaptações do DE.

Neste trabalho, alguns algoritmos bioinspirados (com versões auto-adaptativas) serão avaliados por testes estatísticos não-paramétricos para três aplicações: um *benchmark* da competição CEC2017 e duas em problemas de controle preditivo não-linear (NMPC).

1.1 Objetivos da Dissertação

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é a análise de desempenho de algoritmos bioinspirados recentes (2014 até 2017), clássicos (PSO e DE) e com adaptação (JADE e LSHADE) utilizando para isso testes estatísticos não-paramétricos e, após a análise, realizar modificações e adaptações nos novos algoritmos mais bem ranqueados a fim de melhorar seus desempenhos perante os problemas apresentados. Esses problemas são apresentados como estudos de caso, dos quais estão: dois otimizadores do controle preditivo não-linear (pêndulo e simulação de satélites) e um benchmark da competição CEC2017 com funções mono-objetivas com restrição de fronteira.

1.1.2 Objetivos Específicos

Para alcançar o objetivo geral deste trabalho, os seguintes objetivos específicos foram elaborados:

1. Aplicar os algoritmos nos três estudos de caso e extrair a mediana do desempenho de cada algoritmo/problema;
2. Verificar a preferência do uso dos testes não-paramétricos de desempenho através da aplicação dos testes de normalidade e homogeneidade dos dados;
3. Aplicar os testes não-paramétricos da comparação múltipla de $N \times N$ algoritmos, validando com os procedimentos post-hoc;
4. Realizar análise qualitativa dos dados mediante estados dos problema de controle e resultados das funções da competição CEC2017;
5. Apresentar, como um complemento, a comparação dos tempos de execução de cada algoritmo;
6. Apresentar as modificações e adaptações dos novos algoritmos que tenham mostrado os melhores desempenhos;
7. Mostrar com os testes não-paramétricos a melhora do desempenho desses algoritmos com modificações;

1.2 Contribuições do Manuscrito

Este trabalho se propõe apresentar novas versões de recentes algoritmos bioinspirados que apresentem uma melhora no desempenho quando aplicados a um problema determinado. As ferramentas de suporte para a análise desse desempenho são os testes estatísticos não-paramétricos, do qual o uso em trabalhos científicos vem crescendo nos últimos anos. Assim, este trabalho contribui com outros trabalhos do laboratório LEIA (*Laboratory of Embedded Systems and Integrated Circuits Applications*) que desenvolvem aplicações de controle preditivo usando algoritmos bioinspirados como *solver*.

Em dezembro de 2017 foi apresentado o artigo *Bio-inspired optimization applied to the tuning of model predictive control parameters* no COBEM (Congresso Internacional de Engenharia Mecânica) realizado em Curitiba - PR, cujos resultados foram frutos dos estudos deste trabalho (MESQUITA; QUINTERO; SAMPAIO, 2017).

1.3 Apresentação do Documento

Este trabalho é constituído de 5 capítulos, divididos da seguinte forma:

O Capítulo 2 traz a revisão dos tópicos abordados neste trabalho. Inicia apresentando todos os algoritmos, indicando a inspiração, modelo matemático e o pseudo código dos mesmos. Em seguida expõe os testes de normalidade e homogeneidade da variância dos dados, além das ferramentas de análise estatística não-paramétricas dos algoritmos. Na seção 2.2.8 aborda o *software* utilizado para esses testes. Por se tratar de dois dos três estudos de caso deste trabalho, a metodologia e fundamentos básicos do controle preditivo baseado em modelo (MPC) é apresentado na seção 2.3.

O Capítulo 3 apresenta a metodologia utilizada neste trabalho. Inicia mostrando a forma como os dados de desempenho dos algoritmos serão tratados e indica a sequência de testes necessários para a comparação deles. Na seção 3.5 mostra como é feita a comparação no tempo de execução dos algoritmos.

Os resultados do trabalho são mostrados no Capítulo 4 mostrando os valores dos testes, a análise qualitativa dos dados e a verificação do tempo de execução dos algoritmos. O capítulo finaliza apresentando o desempenho dos novos algoritmos com suas modificações e adaptações.

Por fim no Capítulo 5 são apresentadas as conclusões e as devidas sugestões para trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Este capítulo aborda sobre os temas relevantes do trabalho, trazendo uma revisão dos principais tópicos da teoria pertinente. Na seção 2.1 são apresentadas as meta-heurísticas analisadas, descrevendo suas características e principais aspectos de funcionamento. A comparação do desempenho das meta-heurísticas é feita por meio dos métodos não-paramétricos e a seção 2.2 traz a teoria e os critérios para o seu uso, além de apresentar o software, com todos os testes estatísticos implementados, utilizado nesta dissertação. Por fim, uma breve apresentação acerca dos princípios do Controle Preditivo baseado em Modelo (MPC) é feita na seção 2.3, pois dois dos três estudos de casos apresentados são problemas de controle que utilizam esta metodologia.

2.1 Algoritmos Bioinspirados

Atualmente, pode ser vista muita tecnologia que foi desenvolvida baseada em características e comportamentos encontrados na natureza; neste contexto, o velcro inspirado na textura de algumas plantas e o sonar, baseado no sistema de localização dos morcegos, são exemplos dessa tecnologia.

Em certas áreas de estudo a computação busca inspirar-se em comportamentos de modelos bem sucedidos na natureza, surgindo assim os *algoritmos bioinspirados*. As duas principais classes são: (a) os algoritmos evolucionários e (b) os algoritmos baseados em inteligência coletiva. Aqueles, fixados pela teoria de Darwin, são baseados na evolução dos seres e estes últimos são fruto da teoria de inteligência de enxame.

O termo *enxame* é utilizado de forma genérica para se referir a qualquer coleção estruturada de agentes capazes de interagir entre si. Logo, pode-se dizer que as interações coletivas de todos os agentes, dentro do sistema, muitas vezes levam a algum tipo de comportamento ou inteligência coletiva. Esse tipo de inteligência artificial inclui qualquer tentativa de projetar algoritmos ou dispositivos distribuídos de solução

de problemas sem ter um controle centralizado, inspirado no comportamento coletivo de agentes sociais e outras sociedades de animais e insetos (KAWAI et al., 2007).

No presente trabalho buscou-se analisar alguns dos algoritmos bioinspirados consolidados pela comunidade científica, como o PSO (*Particle Swarm Optimization*) e o DE (*Differential Evolution*) clássicos, que serão descritos nas subseções 2.1.1 e 2.1.2, e também alguns dos mais recentes métodos encontrados na literatura: GWO - *Grey Wolf Optimizer*, MFO - *Moth-Flame Optimization*, DA - *Dragon-Fly Algorithm*, WOA - *Whale Optimization Algorithm*, SCA - *A Sine Cosine Algorithm* - e SSA - *Salp Swarm Algorithm*. Além dessas, outras duas meta-heurística são objetos de estudo: o JaDE e o LSHADE, métodos adaptativos do DE clássico e que serão descritos na subseção 2.1.3.

Antes de descrevê-los, algumas definições serão feitas e usadas em todas as meta-heurística (em termos de nomenclatura), quais sejam:

- (a) S - número de indivíduos da população (agentes de busca)
- (b) D - dimensão do problema a ser resolvido
- (c) t - iteração atual
- (d) Max_{iter} - número máximo de iterações
- (e) lb e ub são os limites inferior e superior do espaço de busca

Além desses, os agentes de busca da população são denominados de X , matriz do tipo $S \times D$,

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & \cdots & X_{1,D} \\ X_{2,1} & X_{2,2} & \cdots & \cdots & X_{2,D} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{S,1} & X_{S,2} & \cdots & \cdots & X_{S,D} \end{bmatrix} \quad (2.1)$$

Assim, cada indivíduo da população é representada por uma linha da matriz X . Para exemplificar $X_{i,j}^t$, é a nomenclatura da j -ésima dimensão do i -ésimo agente no curso da iteração t . A aptidão de cada agente de busca é obtida pela sua avaliação na *função custo* do problema. Desta maneira, partindo de $X_{S \times D}$ tem-se, então, a matriz $F_{S \times 1}$ das aptidões.

$$F = \begin{bmatrix} F(X_1) \\ F(X_2) \\ \vdots \\ F(X_S) \end{bmatrix} \quad (2.2)$$

2.1.1 PSO - *Particle Swarm Optimization*

- (a) **Inspiração Biológica:**

Na década de 80, o comportamento dos pássaros despertou um grande interesse de alguns pesquisadores, dos quais se destacou o biólogo Frank Heppner (HEPPNER; GRENANDER, 1990). Heppner descobriu que pássaros, comumente vistos em bandos, possuem padrões sociais que auxiliam na sobrevivência do grupo. Para procurar comida, por exemplo, eles se espalham e cada indivíduo se torna um explorador utilizando critérios aleatórios. Quando algum deles encontra alimento, a informação é transmitida para todos que aos poucos vão de encontro ao primeiro. Esse conhecimento compartilhado aumenta a probabilidade de alcançar o objetivo e é a base da já mencionada inteligência de enxames. Partindo desse comportamento coletivo dos pássaros o PSO foi criado e seu modelo teórico é apresentado no item a seguir.

(b) **Modelo Matemático:**

James Kennedy e Russel Eberhart, inspirados no comportamento social dos pássaros estudados por Heppner, desenvolveram uma técnica de otimização que veio a ser conhecida como *enxame de partículas* (*Particle Swarm Optimization*- PSO) (EBERHART; KENNEDY, 1995).

O método de otimização por enxame de partículas é um ramo da inteligência artificial que otimiza um problema iterativamente ao tentar melhorar a solução candidata, com respeito a uma dada medida de qualidade. Fazendo uma analogia, o termo *partícula* foi adotado para simbolizar os pássaros e representar as possíveis soluções do problema a serem encontradas. A área sobrevoada pelos pássaros é equivalente ao espaço de busca e o local com comida, ou o ninho, corresponde a encontrar a solução ótima. Para que o bando de pássaros sempre se aproxime do objetivo, ao invés de se perder ou nunca alcançar o alvo, utiliza-se o indicador denominado de aptidão, ou seja, o desempenho das partículas. Para alcançar o alvo focado, sejam os alimentos ou os ninhos, os pássaros fazem uso de suas experiências individuais e da experiência do próprio bando.

O termo *indicador da experiência* ou *conhecimento individual* de cada partícula, isto é, seu histórico de vida, é o *pbest*. O melhor *pbest* é o responsável por representar o conhecimento do enxame como um todo, denominado assim de *gbest*. Além da posição do indivíduo, outra relevante informação das partículas para o PSO é a *velocidade* e o cálculo dela é realizado pela equação a seguir:

$$V_i^{t+1} = V_i^t + c_1 U_1 (pbest_i^t - X_i^t) + c_2 U_2 (gbest^t - X_i^t) \quad (2.3)$$

onde V_i^t é a velocidade da partícula i na iteração t limitada pelo intervalo $[-V_{max}, V_{max}]$, c_1 é o coeficiente cognitivo e c_2 é o coeficiente social que determinam o grau de confiança na partícula e no enxame, respectivamente. Além disso, U_1 e U_2 são números aleatórios no intervalo $[0,1]$.

Obtida a velocidade, o deslocamento da partícula é realizado pela seguinte fórmula:

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2.4)$$

Vale observar que c_1 e c_2 têm papel fundamental no desempenho do PSO, já que valor alto do primeiro torna o valor individual da partícula mais relevante enquanto que maior valor do segundo faz com que a confiança seja transmitida ao enxame.

A versão do PSO utilizada neste trabalho possui uma modificação denominada de *fator de inércia*,

proposta por Shi e Eberhart (1998). Esse fator é aplicado à velocidade da partícula, sendo representado por um peso (w) cujo valor decresce linearmente ao longo das iterações de w_0 a w_f . O alto valor de w faz com que no início da execução da meta-heurística as partículas façam a busca global (fase de exploração- *exploration phase*) e ao diminuir favorece a busca local (fase de refinamento - *exploitation phase*). O cálculo da velocidade com fator de inércia é realizado pela equação seguinte:

$$V_i^{t+1} = w \cdot V_i^t + c_1 U_1(pbest_i^t - X_i^t) + c_2 U_2(gbest^t - X_i^t) \quad (2.5)$$

No próximo item é apresentado o pseudo código do algoritmo PSO.

(c) **Pseudo código:**

Com o modelo obtido no item anterior, pode-se observar o pseudo código do PSO no **Algoritmo 1**.

Algoritmo 1 : PSO

Entrada: S , Max_{iter} , lb e ub

Saída: posição do melhor agente de busca $gbest$ e sua aptidão $F(gbest)$

1: **Início**

2: Inicializa o enxame de partículas $X_i(i = 1, 2, \dots, S)$;

3: Calcula a aptidão de cada agente de busca

4: Determina $pbest$ de cada partícula

5: Seleciona o melhor $pbest$ como $gbest$

6: **repete**

7: **para** cada agente de busca **faça**

8: Calcula a velocidade do agente atual com a Eq. (2.5)

9: Calcula o deslocamento do agente atual com a Eq. (2.4)

10: **fim para**

11: Calcula a aptidão de todos os agentes de busca

12: Atualiza $pbest$ de cada partícula

13: Atualiza $gbest$ se for o caso

14: Atualiza w

15: $t = t + 1$

16: **até** $t \geq Max_{iter}$

17: **fim**

2.1.2 DE - Evolução Diferencial

- (a) **Inspiração Biológica:** DE (*differential evolution* em inglês) é uma meta-heurística proposta por Storn e Price (1997) proveniente da computação evolutiva. Esse ramo de pesquisa inspira-se nas teorias de Darwin sobre a evolução das espécies. Os primeiros passos da área aconteceram com a criação dos algoritmos genéticos por Holland (1973), que introduziu operadores genéticos na otimização de problemas, produzindo um comportamento adaptativo dos agentes em busca do ponto ótimo da função.

Da genética foram extraídos três conceitos importantes: (a) *mutação*, (b) *cruzamento cromossômico* (*Crossing-over* em inglês) e (c) *seleção*. A mutação ocorre quando há mudanças no material genético de um organismo, e a causa pode estar relacionada a erros durante a divisão celular, devido à exposição a radiação ultravioleta ou até mesmo por vírus. O cruzamento cromossômico ocorre com a troca de partes do material genético de um cromossomo para outro. A seleção está dentro da teoria de sobrevivência de Darwin, a qual diz que o indivíduo mais preparado ao meio sobrevive em decorrência do mais fraco. A partir desses três conceitos o modelo matemático do DE possui três operadores, mostrados no item a seguir.

(b) **Modelo matemático:**

A meta-heurística DE segue a estratégia evolutiva de seus agentes de busca e para cada indivíduo X_i são aplicados os três operadores a seguir:

(i) **Mutação:**

A cada iteração t , o operador mutação gera um vetor mutante V_i^t originário de três agentes aleatoriamente escolhidos na população. O cálculo desse vetor é realizado pela equação:

$$V_i^t = X_{r1} + F_m \cdot (X_{r2} - X_{r3}) \quad (2.6)$$

onde $r1$, $r2$ e $r3$ são os índices dos agentes escolhidos aleatoriamente e $r1$ é denominado de pai de V_i . F_m é o fator de mutação o qual tem valor constante de 0,5. Após a criação de V_i , segue-se para o segundo passo, o *crossover*.

(ii) **Crossover:**

Após a mutação, ocorre o processo de cruzamento (*crossover*) das informações do agente mutante e de outros da população. Esse cruzamento gera o agente U_i^t e seu valor é obtido com a seguinte expressão:

$$U_{i,j}^t = \begin{cases} V_{i,j} & \text{se } rand(0,1) \leq CR \text{ || } j = j_{rand} \\ X_{i,j}^t & \text{se caso contrario} \end{cases} \quad (2.7)$$

onde $rand(0,1)$ é uma função que retorna um número uniformemente aleatório entre 0 e 1. CR é a constante de *crossover* com valor de 0,9 e j_{rand} é o índice escolhido aleatoriamente de 1 a D . Esse último parâmetro garante que pelo menos uma das informações do agente mutante seja transferido ao U_i (quando $j = j_{rand}$).

Terminada a etapa do *crossover*, a meta-heurística segue para o último passo, a *seleção*.

(iii) **Seleção:**

A etapa de seleção verifica se o agente U_i proveniente dos processos de mutação e *crossover* é melhor que o agente de índice i (X_i). Dessa forma, a operação de seleção é realizada pela equação (2.8).

$$X_i^{t+1} = \begin{cases} U_i^t & \text{se } F(U_i^t) \leq F(X_i^t) \\ X_i^t & \text{se } F(U_i^t) > F(X_i^t) \end{cases} \quad (2.8)$$

onde $F(X)$ é a avaliação da função objetivo (função custo) a ser minimizada.

A estrutura do DE, apresentada nesta subseção, é denominada de DE/rand/1, devido ao processo de mutação ser baseado em três agentes aleatórios, outros tipos de mutação foram criados e serão descritos na subseção 2.1.3.

- (c) **Pseudo código:** A metodologia de busca do DE utilizando os processos da mutação, *crossover* e seleção é apresentada na forma de pseudo código no **Algoritmo 2**.

Algoritmo 2 : DE

Entrada: S , Max_{iter} , lb e ub

Saída: posição do melhor agente de busca X_{best} e sua aptidão $F(X_{best})$

- 1: **Início**
 - 2: Inicializa a população de agentes $X_i (i = 1, 2, \dots, S)$;
 - 3: Inicializa as constantes F e CR
 - 4: Calcula a aptidão de cada agente de busca
 - 5: $X_{best} =$ o melhor agente de busca
 - 6: **repete**
 - 7: **para** cada agente de busca $i = 1 \dots S$ **faça**
 - 8: **Mutação:**
 - 9: Geração do vetor mutante pela Eq. (2.6).
 - 10: **Crossover:**
 - 11: Realiza o cruzamento de informações com a Eq. (2.7).
 - 12: **Seleção:**
 - 13: Atualiza o agente da próxima geração aplicando a Eq. (2.8).
 - 14: **fim para**
 - 15: Calcula a aptidão de todos os agentes de busca
 - 16: Atualiza X_{best} se for o caso
 - 17: $t = t + 1$
 - 18: **até** $t \geq Max_{iter}$
 - 19: **fim**
-

2.1.3 DE adaptativo

O DE apresentado na subseção 2.1.2 é denominado clássico, pois a partir dele outras versões foram criadas. Essas outras meta-heurísticas diferenciam-se no cálculo do processo de mutação e estabelecem, assim, uma família de DE's. Ela é classificada pela convenção DE/x/y, o qual x representa a escolha do agente base (X_i) e y é o número de diferenças consideradas no cálculo (DAS; SUGANTHAN, 2011). Além do DE/rand/1, outras versões são comumente usadas na literatura (CUI et al., 2016), como:

1. DE/rand/2:

$$V_i^t = X_{r1}^t + F_m \cdot (X_{r2} - X_{r3}) + F_m \cdot (X_{r4} - X_{r5}) \quad (2.9)$$

2. DE/best/1:

$$V_i^t = X_{best}^t + F_m \cdot (X_{r_1} - X_{r_2}) \quad (2.10)$$

3. DE/best/2:

$$V_i^t = X_{best}^t + F_m \cdot (X_{r_1} - X_{r_2}) + F_m \cdot (X_{r_3} - X_{r_4}) \quad (2.11)$$

4. DE/current-to-best/1:

$$V_i^t = X_i^t + F_m \cdot (X_{best}^t - X_i^t) + F_m \cdot (X_{r_1} - X_{r_2}) \quad (2.12)$$

onde r_1, r_2, r_3, r_4 e r_5 são índices escolhidos aleatoriamente de 1 a S e X_{best}^t é o melhor agente na iteração t .

(a) Adaptações dos parâmetros:

Nas meta-heurísticas mostradas até aqui o fator de mutação F_m é um valor constante entre $[0,1]$ para todo o processo de minimização. Além disso, CR também é constante, tendo papel importante no desempenho da meta-heurística. A sintonização do valor desses parâmetros não é uma tarefa simples e para problemas mais complexos é manualmente impraticável. Baseado nisso, diferentes mecanismos de adaptação foram desenvolvidos, a exemplo o FADE que usa controladores de lógica fuzzy e identifica o valor dos operadores de mutação e *crossover* ao longo das iterações (LIU; LAMPINEN, 2005).

Omran, Salman e Engelbrecht (2005) propuseram a auto-adaptação do fator de *crossover* (CR), ajustando-o para cada agente de busca em uma distribuição normal de $\mu = 0,5$ e $\sigma = 0,15$, sendo que essa versão foi denominada SDE. Depois do SDE, em 2009 foi desenvolvida outra versão auto-adaptativa dos parâmetros do DE, dessa vez denominada de SaDE (QIN; HUANG; SUGANTHAN, 2009), na qual são implementadas duas das estratégias de mutação: (a) *DE/rand/1* e (b) *DE/current-to-best/1*.

A escolha da estratégia é feita pela taxa de sucesso de ambas, nas últimas 50 iterações. Além disso, o fator de mutação do SaDE é gerado individualmente para cada indivíduo a cada iteração, de acordo com uma distribuição normal de $\mu = 0,5$ e $\sigma = 0,3$ no intervalo $[0,2]$ pela expressão $N(0,5; 0,3)$. CR_i também é gerado a cada iteração de distribuição normal, porém, com $\mu = CR_m$ e $\sigma = 0,1$ ($N(CR_m; 0,1)$), o qual CR_m inicia com valor 0,5 e é atualizado a cada 5 iterações de acordo com os valores das gerações de sucesso CRs pela expressão $CR_m = median(CRs)$. A probabilidade de escolha da estratégia n (1 ou 2) na iteração t é dada por:

$$p_n^t = \frac{S_n^t}{\sum_{n=1}^2 S_n^t} \quad (2.13)$$

onde S_n representa a razão de sucesso dos vetores gerados pela estratégia n .

(b) JADE - Adaptive Differential Evolution

O foco desse item é apresentar o JADE (ZHANG; SANDERSON, 2009), meta-heurística que implementa uma estratégia de mutação denominada *DE/current-to-pbest* com adaptação de F_m e CR . Ao longo das iterações são armazenados, em um conjunto \mathbf{A} , os pais que geraram indivíduos de sucesso na seleção. O arquivo \mathbf{A} fornece melhores informações sobre a direção pela qual a solução está sendo direcionada durante processo de busca. Dessa forma, em *DE/current-to-pbest* a mutação ocorre de acordo com a seguinte equação:

$$V_i^t = X_i + F_{m_i} \cdot (X_{best}^p - X_i) + F_{m_i} \cdot (X_{r1} - \tilde{X}_{r2}) \quad (2.14)$$

onde X_{best} é escolhido aleatoriamente como um dos $100 \cdot p\%$ melhores dentre a população atual \mathbf{P} com $p \in (0, 1]$ e \tilde{X}_{r2} é escolhido da união de $\mathbf{P} \cup \mathbf{A}$. O *crossover* do JADE é semelhante ao do DE clássico, exceto que na equação (2.7) a constante de operação (CR_i) é particular a cada agente i .

Neste caso, a escolha dos valores de CR_i é realizada com base em uma distribuição normal de média μ_{CR} e desvio padrão 0,1 como mostra a equação (2.15).

$$CR_i = randn_i(\mu_{CR}; 0, 1) \quad (2.15)$$

A média (μ_{CR}) é inicializada com valor de 0,5 e atualizada a cada iteração, como se segue:

$$\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot mean_A(S_{CR}) \quad (2.16)$$

onde c é uma constante com valor de 0,1; S_{CR} é o conjunto que armazena todos os CR_i 's que obtiveram sucesso em interações anteriores e $mean_A(\cdot)$ é a operação de média aritmética usual.

Já para a escolha do fator de mutação (F_{m_i}) usa-se a distribuição de Cauchy com os parâmetros de localização e escala como μ_F e 0,1.

$$F_{m_i} = randc_i(\mu_F; 0, 1) \quad (2.17)$$

onde F_{m_i} está no intervalo $[0,1]$. Assim como S_{CR} , S_F é o conjunto de fatores de mutação que obtiveram sucesso em outras iterações. O parâmetro μ_F é inicializado com o valor de 0,1 e atualizado conforme a equação abaixo:

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot mean_L(S_F) \quad (2.18)$$

onde $mean_L(\cdot)$ é a média de Lehmer, calculada da seguinte forma:

$$mean_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F} \quad (2.19)$$

Assim, os processos de *crossover* e seleção do JADE são realizados pelas equações (2.20) e (2.21).

$$U_{i,j}^t = \begin{cases} V_{i,j} & \text{se } rand(0, 1) \leq CR_i \parallel j = j_{rand} \\ X_{i,j}^t & \text{se caso contrario} \end{cases} \quad (2.20)$$

$$X_i^{t+1} = \begin{cases} U_i^t; A \leftarrow X_i^t; S_{CR} \leftarrow CR_i; S_F \leftarrow F_i & \text{se } F(U_i^t) \leq F(X_i^t) \\ X_i^t & \text{se } F(U_i^t) > F(X_i^t) \end{cases} \quad (2.21)$$

Para finalizar, no **Algoritmo 3** são observadas as etapas do JADE em forma de pseudo código.

Algoritmo 3 : JADE

Entrada: S , Max_{iter} , lb e ub

Saída: posição do melhor agente de busca X_{best} e sua aptidão $F(X_{best})$

- 1: **Início**
 - 2: Inicializa a população de agentes $X_i (i = 1, 2, \dots, S)$;
 - 3: Inicializa as constantes $\mu_{CR} = 0,5$ e $\mu_F = 0,5$
 - 4: Calcula a aptidão de cada agente de busca
 - 5: $X_{best} =$ o melhor agente de busca
 - 6: **repete**
 - 7: **para** cada agente de busca $i = 1 \dots S$ **faça**
 - 8: Gera CR_i pela Eq. (2.15)
 - 9: Gera F_i pela Eq. (2.17)
 - 10: **Mutação:**
 - 11: Geração do vetor mutante pela Eq. (2.14).
 - 12: **Crossover:**
 - 13: Realiza o cruzamento de informações com a Eq. (2.20).
 - 14: **Seleção:**
 - 15: Atualiza o agente da próxima geração aplicando a Eq. (2.21).
 - 16: **fim para**
 - 17: Calcula a aptidão de todos os agentes de busca
 - 18: Atualiza X_{best} se for o caso
 - 19: Atualiza μ_{CR} pela Eq. (2.16)
 - 20: Atualiza μ_F pela Eq. (2.18)
 - 21: $t = t + 1$
 - 22: **até** $t \geq Max_{iter}$
 - 23: **fim**
-

(c) LSHADE - SHADE using Linear Population Size Reduction

Como mostrado anteriormente, JADE é uma versão do DE com adaptação dos parâmetros de mutação e *crossover*. Essa adaptação está diretamente ligada ao armazenamento dos valores de CR_i e F_i , que permitiram a evolução dos indivíduos, nos conjuntos S_{CR} e S_F . Apesar do armazenamento ocorrer apenas na verificação de seleção, não é garantido que valores pobres em informação dos parâmetros não sejam incluídos. A inclusão de valores resultam em μ_{CR} e μ_F indesejáveis, o que compromete o desempenho da meta-heurística.

A fim de tornar mais robusto o comportamento do JADE foi criado o SHADE (*Success-History based Adaptive DE*) (TANABE; FUKUNAGA, 2013). No SHADE foi implementado o histórico de memórias (M_{CR} e M_F) das médias dos valores de S_{CR} e S_F , armazenadas a cada iteração.

As memórias possuem tamanho fixo H e são inicializadas com valor de 0,5. A cada iteração os parâmetros CR_i e F_{m_i} são atualizados conforme as equações (2.22) e (2.23).

$$CR_i = randn_i(M_{CR,r_i}; 0, 1) \quad (2.22)$$

$$F_{m_i} = randc_i(M_{F,r_i}; 0, 1) \quad (2.23)$$

onde r_i é o índice de seleção aleatoriamente escolhido no intervalo $[1, H]$. O armazenamento das médias nas memórias é realizado pelas equações (2.24) e (2.25).

$$M_{CR,k}^{t+1} = \begin{cases} mean_{WA}(S_{CR}) & \text{se } S_{CR} \neq \emptyset \\ M_{CR,k}^t & \text{caso contrario} \end{cases} \quad (2.24)$$

$$M_{F,k}^{t+1} = \begin{cases} mean_{WL}(S_F) & \text{se } S_F \neq \emptyset \\ M_{F,k}^t & \text{caso contrario} \end{cases} \quad (2.25)$$

onde k ($1 \leq k \leq H$) indica a posição da memória que será atualizada. As funções $mean_{WA}(\cdot)$ e $mean_{WL}(\cdot)$ são médias ponderadas aritmética e de Lehmer, respectivamente, calculadas conforme as equações (2.26) e (2.27).

$$mean_{WA}(S_{CR}) = \sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR,k} \quad (2.26)$$

$$mean_{WL}(S_F) = \frac{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}^2}{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}} \quad (2.27)$$

onde $w_k = \frac{\Delta f_k}{\sum_{k=1}^{|S_{CR}|} \Delta f_k}$ e $\Delta f_k = |f(U_k^t) - f(X_i^t)|$. Outra variação no JADE que o SHADE propôs foi o valor de p na mutação $DE/current-to-pbest$. Enquanto que no primeiro p é constante, no segundo cada indivíduo i possui seu p e seu valor é escolhido de acordo com a função aleatória (2.28).

$$p_i = rand \left[\frac{2}{S}; 0, 2 \right] \quad (2.28)$$

A equação (2.28) garante que, no mínimo, dois indivíduos serão escolhidos para determinar o X_{best}^p . Com o objetivo de melhorar ainda mais o desempenho do SHADE, incorporou-se o LPSR (do inglês *Linear Population Size Reduction*), um método de redimensionamento da população que reduz o seu tamanho linearmente ao longo das iterações. Essa nova meta-heurística é denominada L-SHADE (TANABE; FUKUNAGA, 2014). No LPSR, a população inicia com a quantidade de S_{ini} e decresce linearmente até atingir S_{min} , a cada iteração o número de indivíduos é atualizado conforme a fórmula a seguir:

$$S^{t+1} = round \left[\left(\frac{S_{min} - S_{ini}}{MAX_{iter}} \right) \cdot t + S_{ini} \right] \quad (2.29)$$

onde a função $round(\cdot)$ retorna o valor inteiro mais próximo do cálculo entre os parênteses.

O pseudo código do L-SHADE é mostrado no **Algoritmo 4**.

Algoritmo 4 : LSHADE

Entrada: S, MAX_{iter}, lb e ub

Saída: posição do melhor agente de busca X_{best} e sua aptidão $F(X_{best})$

- 1: **Início**
 - 2: Inicializa a população de agentes $X_i (i = 1, 2, \dots, S)$;
 - 3: Calcula a aptidão de cada agente de busca
 - 4: $X_{best} =$ o melhor agente de busca
 - 5: **repete**
 - 6: **para** cada agente de busca $i = 1 \dots S$ **faça**
 - 7: Define r_i no intervalo $[1, H]$
 - 8: Gera CR_i pela Eq. (2.22)
 - 9: Gera F_i pela Eq. (2.23)
 - 10: **Mutação:**
 - 11: Geração do vetor mutante pela Eq. (2.14)
 - 12: **Crossover:**
 - 13: Realiza o cruzamento de informações com a Eq. (2.20)
 - 14: **Seleção:**
 - 15: Atualiza o agente da próxima geração aplicando a Eq. (2.21)
 - 16: **fim para**
 - 17: Atualiza o tamanho da população pela Eq. (2.29)
 - 18: Calcula a aptidão de todos os agentes de busca
 - 19: Atualiza X_{best} se for o caso
 - 20: Atualiza M_{CR} pela Eq. (2.24)
 - 21: Atualiza M_F pela Eq. (2.25)
 - 22: $t = t + 1$
 - 23: **até** $t \geq MAX_{iter}$
 - 24: **fim**
-

2.1.4 GWO - Grey Wolf Optimizer (2014)

(a) **Inspiração Biológica:**

A meta-heurística GWO (MIRJALILI; MIRJALILI; LEWIS, 2014) é inspirado no comportamento da alcateia de lobos-cinzentos (*Canis Lupus*). Esse comportamento é caracterizado pela hierarquia na liderança e pelo mecanismo de caça. O grupo desses mamíferos é composto em média por 5 a 12 indivíduos, os quais possuem funções específicas. A hierarquia é composta por quatro classes, mostradas na Fig. 2.1.

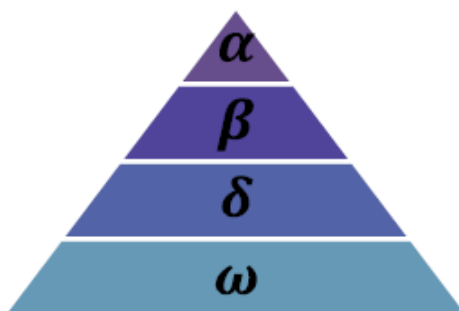


Figura 2.1: Hierarquia entre os lobos-cinzentos (MIRJALILI; MIRJALILI; LEWIS, 2014)

A classe mais alta é o *alpha* (α) composta por um casal de lobos que dominam as demais classes. Por serem os líderes de todo o grupo, são responsáveis por tomar decisões sobre a caça, lugar para dormir, horário para acordar e outras atividades. Além disso, os *alphas* são os únicos permitidos a procriar.

Seguindo a hierarquia, temos a classe *beta* (β) cujos lobos auxiliam os *alphas* na tomada de decisão e ajudam a manter a disciplina do restante do grupo. Em caso de morte de algum *alpha* o melhor candidato a assumir a liderança é algum lobo da classe *beta*.

A terceira classe de dominância é denominada *delta* (δ), os lobos dessa classe são responsáveis por diversas atividades no grupo. Existem os escoteiros que vigiam o perímetro e fazem alertas em caso de perigo, os sentinelas que protegem e buscam manter a segurança do grupo, os anciãos que possuem experiência por já terem sido *alpha* ou *beta* e caçadores que colaboram com as classes mais altas na caça. Há também os que fazem o papel de cuidadores dos lobos mais fracos e doentes. Por fim, a menor classe é a *omega* (ω) que serve em essência como bode-expiatório, mas em alguns casos exerce função de 'babá' dos recém nascidos do grupo.

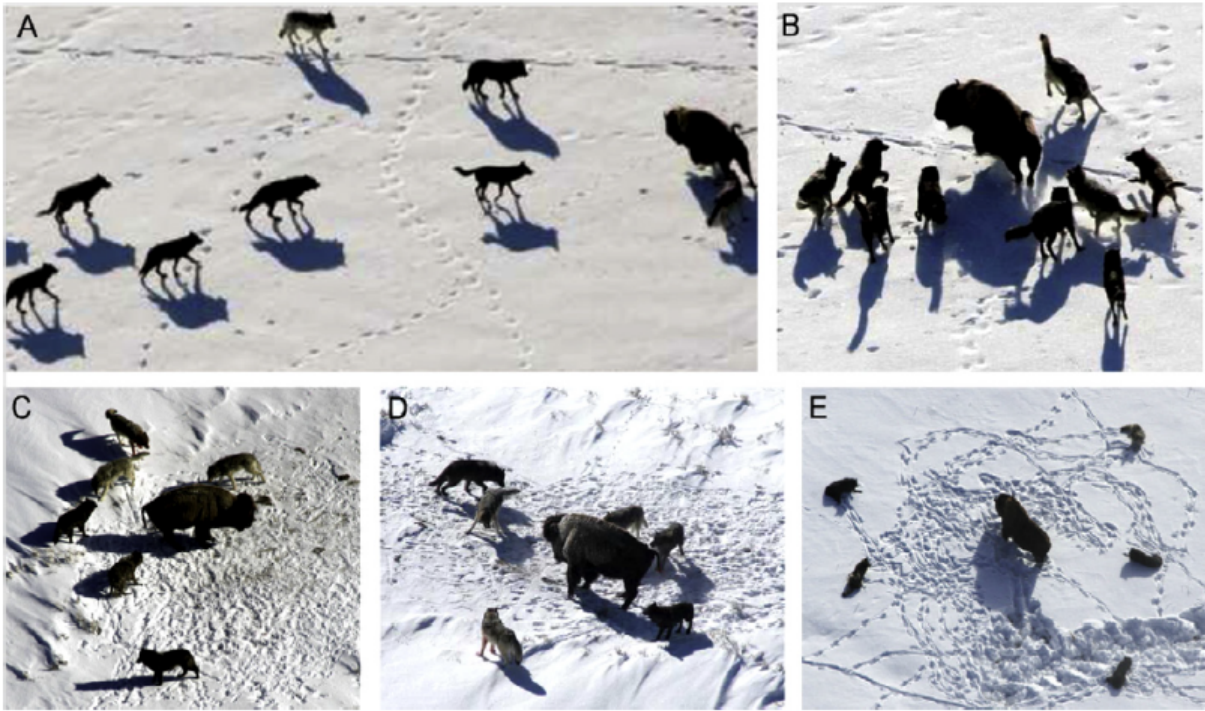


Figura 2.2: Caça dos lobos-cinzentos: A) aproximação, (B-D) cercando a presa e E) aguardando e ataque, (MURO et al., 2011 apud MIRJALILI; MIRJALILI; LEWIS, 2014)

Juntamente com a hierarquia na liderança, os lobos-cinzentos apresentam um mecanismo de caça peculiar que é dividido em três fases principais, observadas na Fig. 2.2. Assim, todo o processo de caça começa com a aproximação da presa, sem deslocamentos bruscos para conservar energia para o ataque, depois os lobos fazem o movimento circular, rodeando a presa a fim de cansá-la, e por fim aguardam que ela não reaja e prossigam para o ataque. No próximo item será abordado como o comportamento dos lobos-cinzentos inspirou o modelo da meta-heurística GWO.

(b) **Modelo Matemático:**

Tendo como base as características sociais e comportamentais dos lobos-cinzentos, o modelamento matemático do GWO é feito conforme os itens a seguir:

- (i) **Definindo a liderança:** Os agentes de busca são divididos em classes iguais aos da Fig. 2.1. *Alpha* (α) é o agente com a melhor solução, à vista disso *beta* (β) e *delta* (δ) são os agentes com a segunda e terceira melhores soluções, respectivamente. Todos os agentes que não forem destas três classes são denominados de *omega* (ω).
- (ii) **Rodeando a presa:** Como mencionado, durante a caça os lobos-cinzentos andam ao redor da presa a fim de confundi-la e evitar que fuja. Quatro equações modelam esse comportamento.

$$D = |C \cdot X_p^t - X_i^t| \quad (2.30a)$$

$$X_i^{t+1} = X_i^t - A \cdot D \quad (2.30b)$$

$$A = 2a \cdot r_1 - a \quad (2.30c)$$

$$C = 2 \cdot r_2 \quad (2.30d)$$

Nas equações (2.30a) e (2.30b), C e A são coeficientes, X_p^t é a posição da melhor solução (presa) até a iteração atual t e X_i^t é a posição do agente de busca i na iteração t . D significa, então, a distância entre X_i e a presa X_p . Os vetores de coeficientes, C e A , são calculados pelas equações (2.30c) e (2.30d) onde a é uma variável que decresce linearmente no decorrer das iterações de 2 a 0, já r_1 e r_2 são números aleatórios no intervalo de 0 a 1.

- (iii) **Caça:** Na alcateia de lobos-cinzentos, a caça acontece em um processo de reconhecimento e rodeamento da presa. Como em um espaço de busca o ponto ótimo é desconhecido, valem-se das três melhores soluções - *alpha*, *beta* e *delta* - para atualizar as posições dos agentes de busca, inclusive dos *omegas*. Dessa forma, baseada nas equações de (2.30a) a (2.30d), a caça foi modelada pelas seguintes fórmulas:

$$D_\alpha = |C_1 \cdot X_\alpha - X_i^t|, D_\beta = |C_2 \cdot X_\beta - X_i^t|, D_\delta = |C_3 \cdot X_\delta - X_i^t| \quad (2.31a)$$

$$X_1 = X_\alpha - A_1 \cdot (D_\alpha), X_2 = X_\beta - A_2 \cdot (D_\beta), X_3 = X_\delta - A_3 \cdot (D_\delta) \quad (2.31b)$$

$$X_i^{t+1} = \frac{X_1 + X_2 + X_3}{3} \quad (2.31c)$$

X_α , X_β e X_δ são as posições das três melhores soluções e D_α , D_β e D_δ representa a distância entre elas e o agente de busca X_i , respectivamente. A_n e C_n ($n = 1, 2, 3$) são calculados pelas equações (2.31c) e (2.31d). Através das equações apresentadas acima, defini-se o deslocamento na iteração $t + 1$ do agente de busca X_i .

- (iv) **Ataque e procura por presas:** O modelo mostrado nas equações (2.31a) a (2.31c) torna a triangulação das posições dos líderes como a região de maior interesse de busca. Isso poderia apresentar para o GWO convergência prematura em mínimos locais, o que é indesejável. Porém, os componentes A e C dispõem papéis fundamentais para que isso não ocorra.

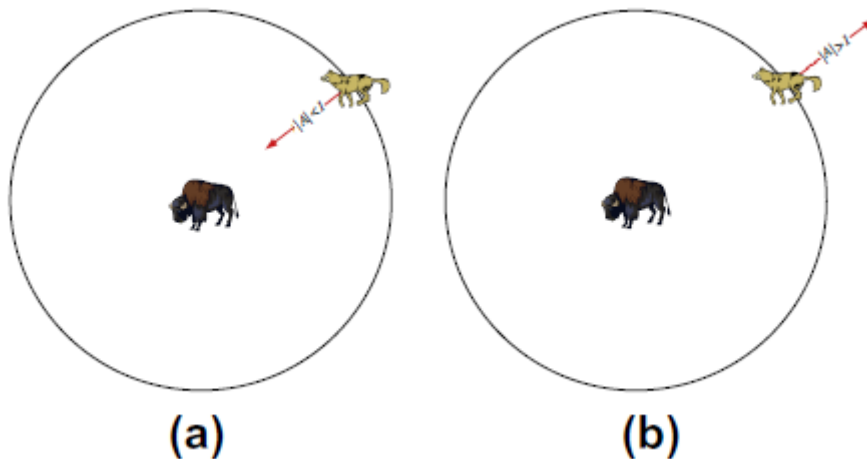


Figura 2.3: (a) refinamento e (b) exploração da busca, adaptado de (MIRJALILI; MIRJALILI; LEWIS, 2014)

Foi visto que a varia de 2 a 0 ao longo das iterações, assim, A possui valores aleatórios no intervalo $[-a, a]$. Para $|A| > 1$, os agentes de busca tendem a se afastar da melhor solução, estabelecendo assim um processo de exploração (busca por presas). Entretanto, quando $|A| < 1$, o processo é inverso, os agentes de busca tendem a se aproximar da melhor solução trazendo um refinamento a busca. Esse comportamento é ilustrado pela Fig. 2.3. Além disso, C também previne a estagnação da solução em ótimos locais.

Na equação (2.30d) é mostrado que C varia aleatoriamente no intervalo de 0 a 2. Assim, esse componente faz com que a busca dê menor ($C < 1$) ou maior ($C > 1$) relevância à posição da presa não só no início, mas em qualquer iteração.

(c) **Pseudo código:**

Para sintetizar a metodologia de busca do GWO, no **Algoritmo 5** é mostrado seu pseudo código.

Algoritmo 5 : GWO

Entrada: S, Max_{iter}, lb e ub

Saída: posição do melhor agente de busca X_α e sua aptidão $F(X_\alpha)$

- 1: **Início**
 - 2: Inicializa a população de lobos-cinzentos $X_i (i = 1, 2, \dots, S)$;
 - 3: Inicializa a, A e C
 - 4: Calcula a aptidão de cada agente de busca
 - 5: $X_\alpha =$ o melhor agente de busca
 - 6: $X_\beta =$ o segundo melhor agente de busca
 - 7: $X_\delta =$ o terceiro melhor agente de busca
 - 8: **repete**
 - 9: **para** cada agente de busca **faça**
 - 10: Atualiza a posição do agente atual com a Eq. (2.31c)
 - 11: **fim para**
 - 12: Atualiza a, A e C
 - 13: Calcula a aptidão de todos os agentes de busca
 - 14: Atualiza X_α, X_β e X_δ
 - 15: $t = t + 1$
 - 16: **até** $t \geq Max_{iter}$
 - 17: **fim**
-

2.1.5 MFO - *Moth-Flame Optimization* (2015)

(a) **Inspiração Biológica:**

As mariposas (*Moth* em inglês) são insetos semelhantes às borboletas, e por serem encontradas essencialmente à noite são conhecidas como borboletas noturnas. Uma de suas características mais interessante é o método de locomoção orientado pela luz da lua, denominado como orientação transversal (Fig. 2.4a). A grande distância entre a Terra e a Lua faz com que a mariposa voe em direção constante, já que mesmo percorrendo vários quilômetros no nosso planeta a Lua permanecerá na mesma direção.

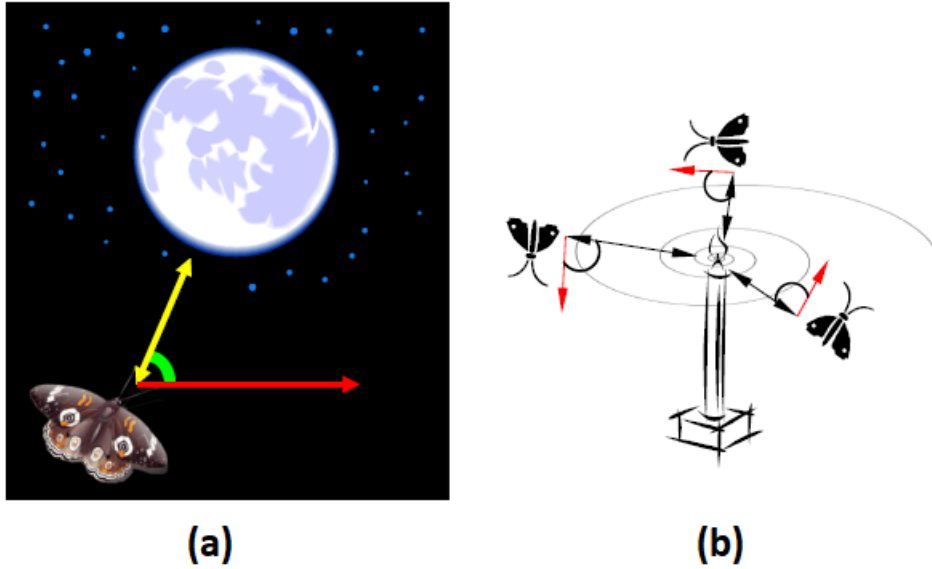


Figura 2.4: Orientação transversal (a) e (b) da busca, *adaptado de* (MIRJALILI, 2015)

Contudo, não temos apenas a Lua como fonte de iluminação. Velas, lamparinas e lâmpadas são fontes artificiais de luz que também atraem as mariposas e independentemente da referência o método de locomoção por orientação é o mesmo. Porém, nesse caso, por estarem próximas à luz o aspecto desse deslocamento é na forma de uma espiral, observada na Fig. 2.4b. Esse movimento ocasionalmente resulta no encontro da mariposa com a luz e essa convergência é objeto de estudo da meta-heurística MFO proposto por Mirjalili (2015).

(b) **Modelo matemático:**

No MFO, são definidas duas variáveis principais, as mariposas (agentes de busca) já denominadas de X e as chamadas de C_h . C_h são as posições das mariposas que possuem a melhor aptidão até a iteração atual, ou seja, são pontos do espaço de busca que tendem a atrair os agentes de busca. Essa atração, como dito anteriormente, é baseada em um movimento espiral, apresentado pela seguinte fórmula:

$$X_i^{t+1} = D_i \cdot e^{br} \cdot \cos(2\pi t) + C_{h_j} \quad (2.32)$$

onde b é uma constante de valor 2, r é um valor aleatório no intervalo $[-1,1]$ e D_i é a diferença entre a j -ésima chama e o i -ésimo agente de busca, calculada pela equação (2.33). Na Fig. 2.5, pode ser observado o possível deslocamento da mariposa em relação ao valor de r , para $r = -1$ o agente de busca fica o mais próximo da chama possível, já $r = 1$ o agente praticamente permanece em seu lugar.

$$D_i = |C_{h_j} - X_i| \quad (2.33)$$

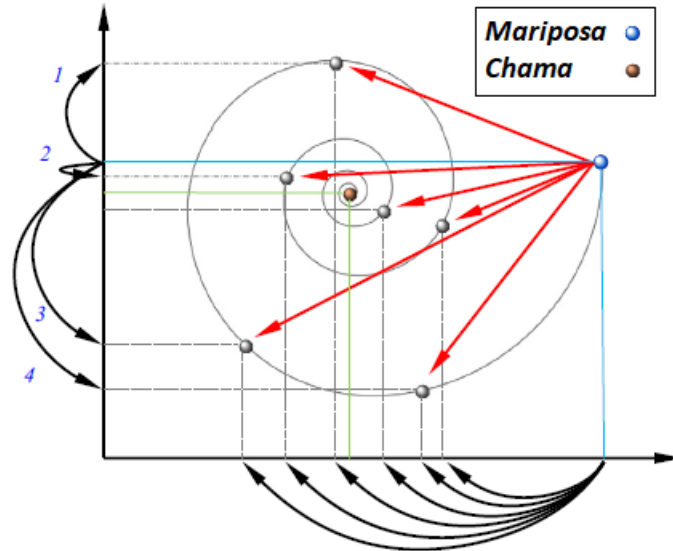


Figura 2.5: Deslocamento em espiral de um agente de busca, *adaptado de* (MIRJALILI, 2015)

Note que r regula a exploração e o refinamento da busca, visto que define o quão próximo da chama o agente de busca irá ficar na próxima iteração. Para dar ênfase na fase de refinamento, quando o número de iterações estiver próximo do fim, r assumirá o seguinte intervalo $[c,1]$, c decresce de -1 a -2 ao longo das iterações e é chamado de constante de convergência.

Para que não ocorra convergência prematura da meta-heurística em mínimos locais, a relação entre uma chama e a mariposa que será atraída inicia como 1:1. Ou seja, não existe a hipótese de todas as mariposas serem atraídas por apenas uma chama. Outra preocupação é se caso o número de chamadas permanecesse constante não haveria o refinamento da solução, já que muitas regiões estarão sendo exploradas. Para resolver esse problema, o número de chamadas será atualizado de acordo com a seguinte fórmula:

$$\#C_h = \text{round} \left(S - t * \frac{S - 1}{Max_{iter}} \right) \quad (2.34)$$

A função $\text{round}(\cdot)$ retorna o inteiro mais próximo do valor calculado entre parênteses.

(c) **Pseudo código:**

Definido o modelo matemático, segue no **Algoritmo 6** o pseudo código do MFO.

Algoritmo 6 : MFO

Entrada: S , Max_{iter} , lb e ub

Saída: posição do melhor agente de busca X_1 e sua aptidão $F(X_1)$

- 1: **Início**
- 2: Inicializa a população de mariposas $X_i (i = 1, 2, \dots, S)$;
- 3: **repete**
- 4: Calcula a aptidão de cada agente de busca
- 5: Ordena a população de mariposas de acordo com a aptidão

- 6: Atualiza o número de chamadas através da Eq. (2.34)
 - 7: Mapeia as chamadas através das mariposas com as melhores aptidões
 - 8: **para** cada agente de busca **faça**
 - 9: Atualiza r
 - 10: Calcular D de acordo com a Eq. (2.33)
 - 11: Atualiza a posição da mariposa de acordo com a Eq. (2.32)
 - 12: **fim para**
 - 13: $t = t + 1$
 - 14: **até** $t \geq Max_{iter}$
 - 15: **fim**
-

2.1.6 DA - *Dragon-Fly Algorithm* (2016)

(a) **Inspiração biológica:**

A libélula (*Dragonfly* em inglês) conhecida no Brasil como 'lavadeira' é um inseto da ordem *Odonata* com milhares de espécies existentes. O ciclo de vida da libélula é dividida em três fases: *ovo*, *ninfa* e *adulta*. Ninfa é a fase posterior à eclosão do ovo e seu período é todo embaixo d'água, além disso é a maior fase da vida desse inseto. Na fase adulta, a ninfa sai de um casulo e adquire asas. Tanto na fase de ninfa quanto na adulta as libélulas são consideradas predadoras, seja na primeira de insetos aquáticos e peixes pequenos seja na segunda de invertebrados terrestres e voadores.

Uma característica importante em relação às libélulas é o seu comportamento coletivo na fase adulta, para alimentação (estático) e migração (dinâmico). No comportamento estático, pequenos grupos sobrevoam determinadas áreas em busca de presas, tais como borboletas e mosquitos (WIKELSKI et al., 2006 apud MIRJALILI, 2016a). Já no comportamento dinâmico, um número maior de libélulas voa em uma única direção a fim de percorrer grandes distâncias (RUSSELL et al., 1998 apud MIRJALILI, 2016a). A Fig. 2.6 ilustra melhor esses dois comportamentos.

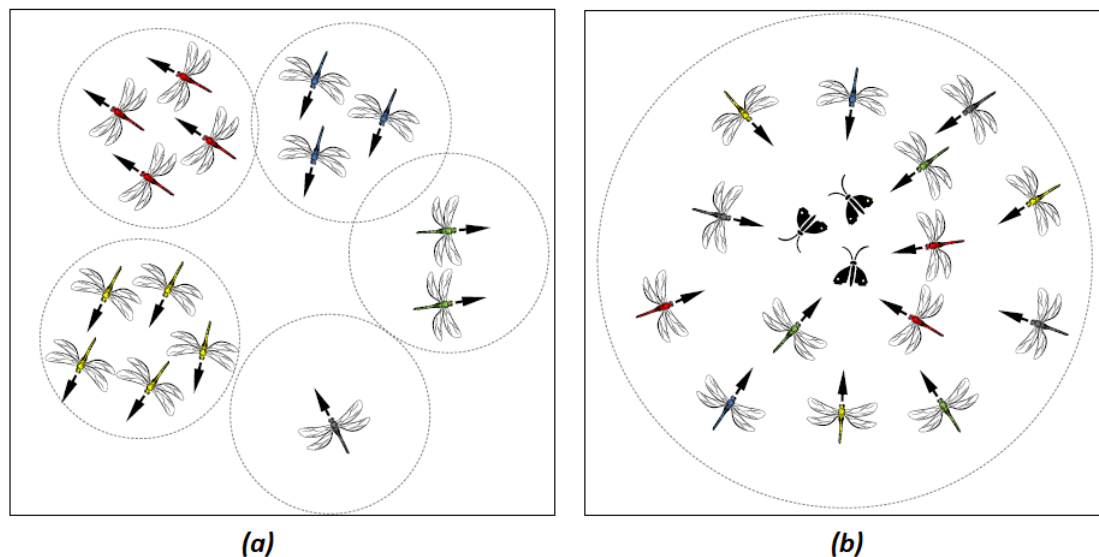


Figura 2.6: Deslocamento estático (a) e dinâmico (b), (MIRJALILI, 2016a)

Esse comportamento coletivo das libélulas inspirou a criação da meta-heurística DA por Mirjalili (2016a), devido à semelhança entre as características estática e dinâmica e as fases de exploração e refinamento de algoritmos meta-heurísticos. A seguir é detalhado a modelagem de todo a meta-heurística DA.

(b) **Modelo matemático:**

Existem algumas variações, dentro do comportamento coletivo dos insetos, que precisam ser levadas em conta para o modelo matemático da meta-heurística DA. Dois objetivos que levam animais e insetos a viverem em grupos são a busca por alimento e afugentar inimigos. Para as libélulas podemos citar cinco padrões de deslocamento individual dentro do grupo. Esses atos podem ser vistos na Fig. 2.7.

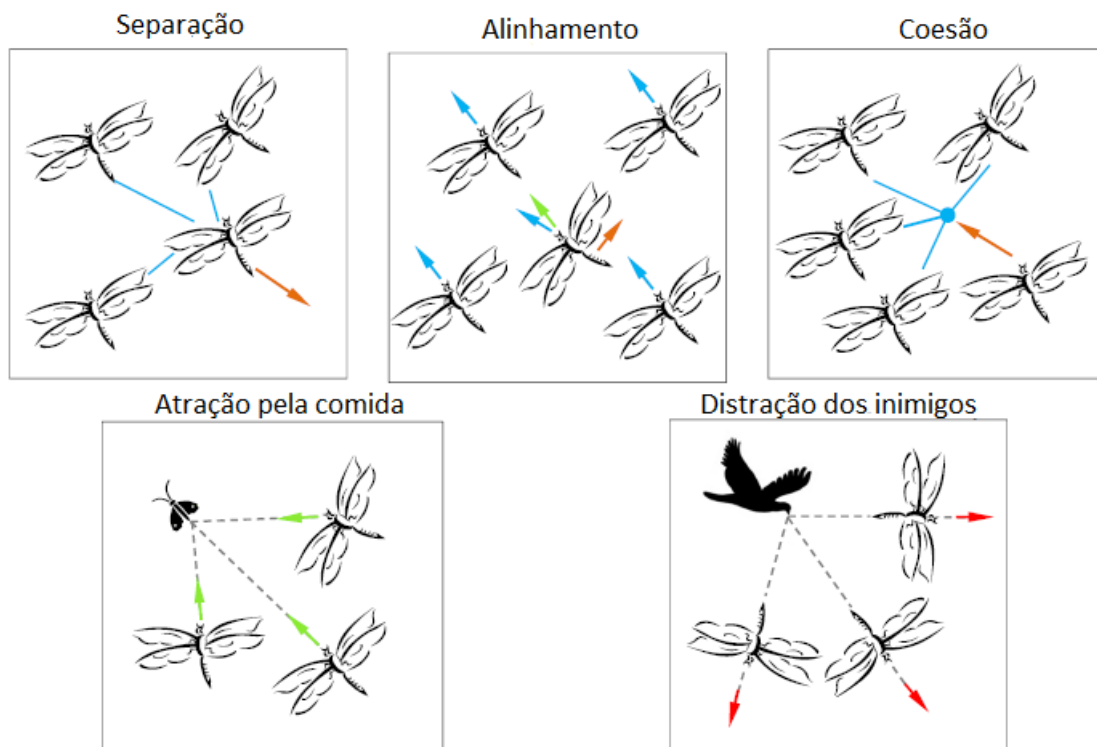


Figura 2.7: Padrões de deslocamento individual em um grupo de libélulas, *adaptado de* (MIRJALILI, 2016a)

Cada um desses padrões de deslocamento foi modelado e as fórmulas são as que seguem abaixo.

(i) **Separação:**

$$SE_i = - \sum_{j=1}^S (X_i - X_j) \quad (2.35)$$

onde i é o índice do atual agente de busca, j é j -ésimo agente de busca vizinho. Vizinho é definido como um agente que se encontra a uma distância dentro de um raio específico, mais a frente será abordada a sua quantificação.

(ii) **Alinhamento:**

$$AL_i = \frac{\sum_{j=1}^S V_j}{S} \quad (2.36)$$

onde V_j representa a velocidade do j -ésimo agente de busca vizinho e que é calculada de iterações sequentes, ou seja, por $V_j = X_j^{t+1} - X_j^t$.

(iii) **Coesão:**

$$CO_i = \frac{\sum_{j=1}^S X_j}{S} - X_i \quad (2.37)$$

onde i é o índice do atual agente de busca, j é o j -ésimo agente de busca vizinho.

(iv) **Atração pela comida:**

$$FS_i = X^+ - X_i \quad (2.38)$$

onde X^+ é a posição do melhor agente de busca até a iteração atual.

(v) **Distração dos inimigos:**

$$DE_i = X^- + X_i \quad (2.39)$$

onde X^- é a posição do pior agente de busca até a iteração atual.

O deslocamento total de um agente de busca é então a combinação desses cinco padrões. Para um indivíduo de índice i , temos:

$$\Delta X_i^{t+1} = (sSE_i + aAL_i + cCO_i + fFS_i + dDE_i) + w\Delta X_i^t \quad (2.40)$$

onde s representa o peso da separação, a o peso do alinhamento, c o peso da coesão, f o peso da atração pela comida e d o peso da distração do inimigo, além desses, w representa o peso inercial e seu valor decresce de 0,9 a 0,4 linearmente ao longo das iterações. Os valores de s , a e c são aleatórios e iguais a seguintes expressão $2 \cdot rand(0, 1) \cdot F_d$, o qual F_d é um fator decrescente de 0,1 a 0. Os outros parâmetros são calculados como: $f = 2 * rand(0, 1)$ e $e = F_d$. ΔX_i^{t+1} é o passo de deslocamento que o agente de busca i percorrerá na iteração $t + 1$, aplicando a seguinte fórmula:

$$X_i^{t+1} = X_i^t + \Delta X_i^{t+1} \quad (2.41)$$

O fator de suma importância nessa meta-heurística é o valor dos pesos apresentado acima. Valores altos dos pesos de alinhamento e de coesão em detrimento dos outros levam à exploração do espaço de busca, já baixos valores desses pesos levam a um refinamento da busca.

Uma solução é a sintonização dos pesos durante a otimização. A transição entre as fases de exploração e refinamento pode ser obtida também tornando o raio de vizinhança dos agentes de busca variável, aumentando proporcionalmente ao número de iterações decorridos. Se em um determinado

valor de raio não houver vizinho de algum indivíduo i , a posição desse agente de busca seguirá a função aleatória denominada de voo de Lévy, conforme a equação a seguir:

$$X_i^{t+1} = X_i^t + Levy(D) \cdot X_i^t \quad (2.42)$$

A função Lévy(x) é explicada em (YANG, 2010 apud MIRJALILI, 2016a). Partindo da modelagem apresentada neste tópico, a seguir encontra-se o pseudo código da meta-heurística DA.

(c) **Pseudo código:**

No **Algoritmo 7** está detalhado o pseudo código do DA.

Algoritmo 7 : DA

Entrada: S , Max_{iter} , lb e ub

Saída: posição do melhor agente de busca X^+ e sua aptidão $F(X^+)$

- 1: **Início**
 - 2: Inicializa a população de libélulas $X_i (i = 1, 2, \dots, S)$;
 - 3: **repete**
 - 4: **para** cada agente de busca **faça**
 - 5: Calcula a aptidão de todos os agentes de busca
 - 6: Atualiza X^+ (comida) e X^- (inimigo)
 - 7: Atualiza w , s , a , f e d
 - 8: Calcula SE, AL, CO, FS e DE com as equações de (2.35) a (2.39)
 - 9: Atualiza o raio da vizinhança
 - 10: **se** uma libélula tem ao menos um vizinho **então**
 - 11: Atualiza a posição do agente de busca com as equações (2.40) e (2.41)
 - 12: **senão**
 - 13: Atualiza a posição do agente de busca com a Eq. (2.42)
 - 14: **fim se**
 - 15: **fim para**
 - 16: $t = t + 1$
 - 17: **até** $t \geq Max_{iter}$
 - 18: **fim**
-

2.1.7 WOA - *Whale Optimization Algorithm* (2016)

(a) **Inspiração biológica:**

Baleias são mamíferos extremamente inteligentes cujos cérebros, em certos aspectos, se assemelham aos dos humanos. Capazes de pensar, julgar, comunicar e até mesmo ter emoções, esses animais despertam a curiosidade de muitos pesquisadores. Há diversas espécies de baleias ao redor do mundo e uma das mais estudadas é a baleia Jubarte (*Megaptera novaeangliae*). A preservação das Jubartes é uma iniciativa tão importante que surgiu no Brasil o Projeto Baleia Jubarte ¹ e para sua divulgação, foi criado o Espaço Baleia Jubarte, localizado na Praia do Forte no Estado da Bahia.

¹www.baleiajubarte.org.br

O que tornou esse animal uma inspiração na elaboração do WOA (MIRJALILI; LEWIS, 2016) foi o seu mecanismo de caça. A alimentação das Jubartes é baseada em pequenos crustáceos e variados tipos de peixes. Uma estratégia eficiente é abocanhar um cardume inteiro e para isso elas se valem de uma técnica denominada de rede de bolhas, observada na Fig. 2.8. A rede de bolhas serve para encurralar o cardume e forçá-lo a subir e concentrar-se na superfície. Essa formação de bolhas serve também para esconder as jubartes até que o ataque aconteça.

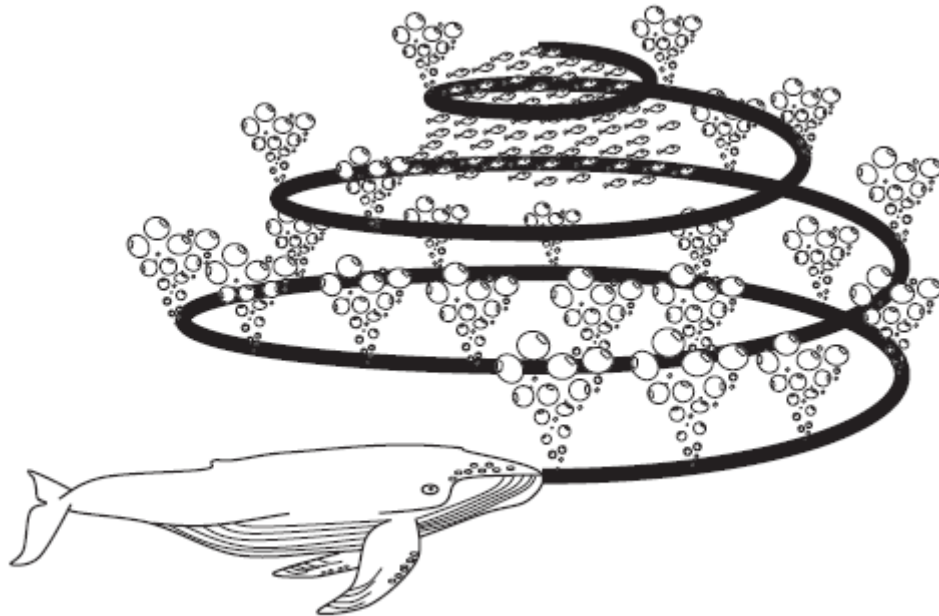


Figura 2.8: Rede de bolhas da baleia Jubarte, (MIRJALILI; LEWIS, 2016)

(b) **Modelo matemático:**

Baseado no comportamento de caça das baleias Jubarte, o WOA foi modelado tendo em conta os seguintes itens:

- (i) **Procura pela presa (exploração):** Esta etapa força os agentes de busca a se espalharem pelo espaço de investigação tendo como referência outro agente escolhido aleatoriamente, independentemente de sua aptidão. As equações que representam essa fase são mostradas abaixo.

$$D_i = |C \cdot X_{aleatorio} - X_i| \quad (2.43a)$$

$$X_i^{t+1} = X_{aleatorio} - A \cdot D_i, \quad (2.43b)$$

$$A = 2a \cdot r - a \quad (2.43c)$$

$$C = 2 \cdot r \quad (2.43d)$$

onde $X_{aleatorio}$ é um agente escolhido aleatoriamente no grupo, r é uma número aleatório no intervalo $[0,1]$ e a é uma variável que decresce linearmente no decorrer das iterações de 2 a 0. Note que o conjunto de equações (2.43) é semelhante ao do GWO nas equações de (2.30a) a

(2.30b), dessa forma, a , A e C possuem a mesma função no WOA. A diferença neste para o GWO é que A e C dependem da mesma variável aleatória r o que faz com que este influencie no deslocamento em relação ao agente de referencia e no peso de sua localização.

Como foi visto na seção 2.1.4, A varia no intervalo $[-a, a]$, a fase de exploração ocorre quando $|A| > 1$. Para $|A| < 1$, outra fase é produzida e pode ser vista no item a seguir.

- (ii) **Ataque por rede de bolhas (refinamento):** Nesta fase são apresentadas duas abordagens distintas. A primeira acontece quando $|A| < 1$ e é determinada pelas fórmulas abaixo, iguais às equações (2.30a) e (2.30b) do GWO.

$$D = |C \cdot X_p - X_i^t| \quad (2.44a)$$

$$X_i^{t+1} = X_p - A \cdot D \quad (2.44b)$$

onde X_p é a posição do agente com a melhor aptidão, A e C são calculados pelas equações do item anterior. A segunda abordagem representa o movimento em espiral que ocorre na formação das redes de bolhas. Primeiramente é calculada a distância entre a posição do agente e a posição da presa (melhor solução), denominada de $D' = |X_p^t - X_i^t|$. Assim o deslocamento em espiral do agente de busca é calculado pela seguinte equação:

$$X_i^{t+1} = D' \cdot e^{bl} \cdot \cos(2\pi l) + X_p^t \quad (2.45)$$

onde $b = 0,5$ e l é um número aleatório no intervalo $[-1,1]$. As Jubartes se locomovem até os cardumes utilizando as duas abordagens conjuntas. Para representar a aleatoriedade na escolha de qual abordagem escolher, o seguinte modelo para a fase do refinamento do WOA foi obtido como:

$$X_i^{t+1} = \begin{cases} X_p^t - A \cdot D & \text{se } p < 0.5 \\ D' \cdot e^{bl} \cdot \cos(2\pi l) + X_p^t & \text{se } p \geq 0.5 \end{cases} \quad (2.46)$$

onde p é um valor aleatório no intervalo $[0,1]$.

(c) **Pseudo código:**

Para verificar a função de cada parte do modelo matemático, no **Algoritmo 8** é mostrado o pseudo código do WOA.

Algoritmo 8 : WOA

Entrada: S , Max_{iter} , lb e ub

Saída: posição do melhor agente de busca X_p e sua aptidão $F(X_p)$

- 1: **Início**
- 2: Inicializa a população de baleias jubarte $X_i (i = 1, 2, \dots, S)$;
- 3: Calcula a aptidão de cada agente de busca
- 4: $X_p =$ o melhor agente de busca
- 5: **repete**

```

6:   para cada agente de busca faça
7:     Atualiza  $a$ ,  $A$ ,  $C$  e  $p$ 
8:     se  $p < 0,5$  então
9:       se  $|A| < 1$  então
10:        Atualiza a posição do agente de busca pela Eq. (2.44b)
11:       senão se  $|A| \geq 1$  então
12:        Seleciona um agente aleatório ( $X_{aleatorio}$ )
13:        Atualiza a posição do agente de busca pela Eq. (2.43b)
14:       fim se
15:     senão se  $p \geq 0,5$  então
16:      Atualiza a posição do agente de busca pela Eq. (2.45)
17:     fim se
18:   fim para
19:   Calcula a aptidão de todos os agentes de busca
20:   Atualiza  $X_p$  se existir melhor solução
21:    $t = t + 1$ 
22: até  $t \geq Max_{iter}$ 
23: fim

```

2.1.8 SCA - A Sine Cosine Algorithm (2016)

(a) Inspiração:

Como já foi dito, há duas fases principais em uma meta-heurística durante a busca pelo mínimo global de uma função: fase de exploração e de refinamento. A meta-heurística que consiga balancear essas duas fases de modo a não estagnar em mínimos locais será sem dúvida uma ferramenta útil na resolução de problemas complexos. Todas as meta-heurísticas apresentadas nas subseções anteriores possuem, além da inspiração biológica, artifícios que implementam a aleatoriedade da natureza, a qual auxilia na ramificação de comportamento da meta-heurística entre as fases de busca, tornando-o mais versátil à otimização do problema.

A busca por uma função que pudesse atingir valores polares, e permitisse a variação do sentido da busca entre as fases, levou Mirjalili (2016b) a elaborar a meta-heurística SCA. Neste trabalho é o único algoritmo não inspirado no comportamento dos seres mas nas funções seno e cosseno. A forma como o SCA foi elaborado é descrita no item a seguir.

(b) Modelo matemático:

O modelo matemático do SCA é baseado então em duas fórmulas: a primeira com a função seno (Eq. 2.47a) e a segunda com a função cosseno (Eq. 2.47b).

$$X_i^{t+1} = X_i^t + r_1 \cdot \text{sen}(r_2) \cdot |r_3 X_p^t - X_i^t| \quad (2.47a)$$

$$X_i^{t+1} = X_i^t + r_1 \cdot \text{cos}(r_2) \cdot |r_3 X_p^t - X_i^t| \quad (2.47b)$$

onde r_1, r_2 e r_3 são números aleatórios e X_p^t é a posição da solução na iteração t . A definição de qual das duas equações seguir é feita por outra variável aleatória r_4 de valores pertencentes ao intervalo $[0,1]$. Assim a combinação das equações (2.47)a e (2.47)b é mostrada na equação (2.48).

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \cdot \text{sen}(r_2) \cdot |r_3 X_p^t - X_i^t| & \text{se } r_4 < 0,5 \\ X_i^t + r_1 \cdot \text{cos}(r_2) \cdot |r_3 X_p^t - X_i^t| & \text{se } r_4 \geq 0,5 \end{cases} \quad (2.48)$$

Os números aleatórios do modelo da meta-heurística possuem funções específicas, r_1 é responsável pelo sentido da movimentação do agente de busca, como observado na Fig. 2.9, r_2 define a velocidade desse movimento sendo maior quanto maior for o módulo de $\text{sen}(r_2)$ ou de $\text{cos}(r_2)$, já r_3 define o peso que a solução terá no deslocamento, se $r_3 > 1$ enfatiza sua posição e $r_3 < 1$ o torna menos relevante no cálculo.

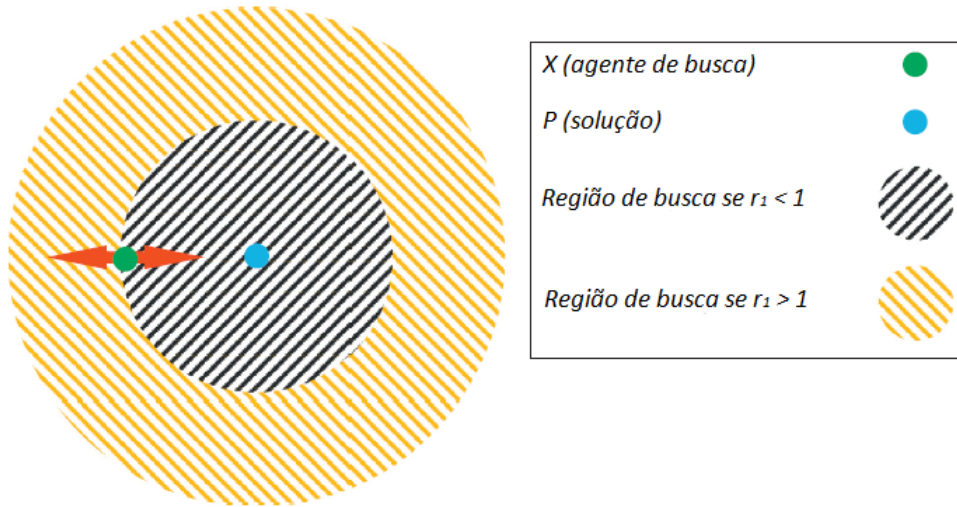


Figura 2.9: Variação da região de busca conforme valor de r_1 , adaptado de (MIRJALILI, 2016b)

O uso das funções seno e cosseno no modelo permite que o movimento tenha um comportamento periódico, fazendo com que o agente de busca possa se aproximar de outra possível solução. Isso dificulta a estagnação da meta-heurística nos mínimos locais. Além disso, a fim de estabelecer um balanceamento entre as fases de exploração e refinamento r_1 é uma variável adaptativa, diminuindo seu valor conforme o curso das iterações. Essa adaptação é realizada pela equação (2.49). Portanto, r_1 permite que SCA inicie a busca na fase de exploração e refine a solução próximo ao número máximo de iterações (Fig. 2.10).

$$r_1 = a - t \cdot \frac{a}{Max_{iter}} \quad (2.49)$$

onde a é um valor constante de 2.

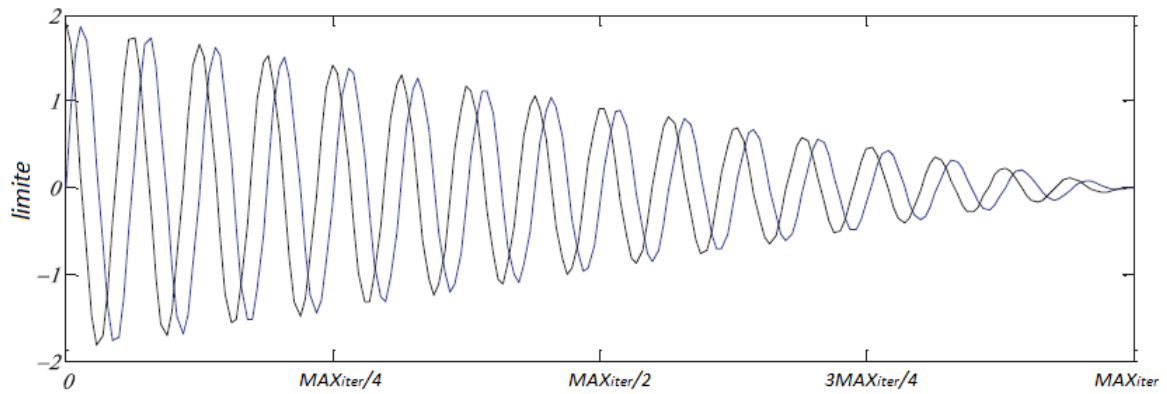


Figura 2.10: Balanceamento entre exploração e refinamento no SCA, adaptado de (MIRJALILI, 2016b)

Definidas as equações e conceitos pertinentes ao SCA, no próximo item é apresentado o pseudo código do algoritmo.

(c) **Pseudo código:**

Tendo em conta o modelo matemático estabelecido no item anterior, no **Algoritmo 9** é mostrado o pseudo código da meta-heurística SCA.

Algoritmo 9 : SCA

Entrada: S , Max_{iter} , lb e ub

Saída: posição do melhor agente de busca X_p e sua aptidão $F(X_p)$

- 1: **Início**
 - 2: Inicializa a posição dos agentes de busca $X_i (i = 1, 2, \dots, S)$;
 - 3: Calcula a aptidão de cada agente de busca
 - 4: $X_p =$ o melhor agente de busca
 - 5: **repete**
 - 6: **para** cada agente de busca **faça**
 - 7: Atualiza r_2 , r_3 e r_4
 - 8: Atualiza r_1 pela Eq. (2.49)
 - 9: **se** $r_4 < 0,5$ **então**
 - 10: Atualiza a posição do agente de busca pela Eq. (2.47a)
 - 11: **senão se** $r_4 > 0,5$ **então**
 - 12: Atualiza a posição do agente de busca pela Eq. (2.47b)
 - 13: **fim se**
 - 14: **fim para**
 - 15: Calcula a aptidão de todos os agentes de busca
 - 16: Atualiza X_p se existir melhor solução
 - 17: $t = t + 1$
 - 18: **até** $t \geq Max_{iter}$
 - 19: **fim**
-

2.1.9 SSA - *Salp Swarm Algorithm* (2017)

(a) **Inspiração biológica:**

Salpas são animais da família da *Salpidae* e vivem no mar em grandes profundidades. Seu corpo tem um formato de barril transparente (Fig. 2.11a) e sua locomoção é similar ao das águas vivas, pelo bombeamento da água através do seu corpo, o que auxilia na alimentação baseada na ingestão de plânctons.

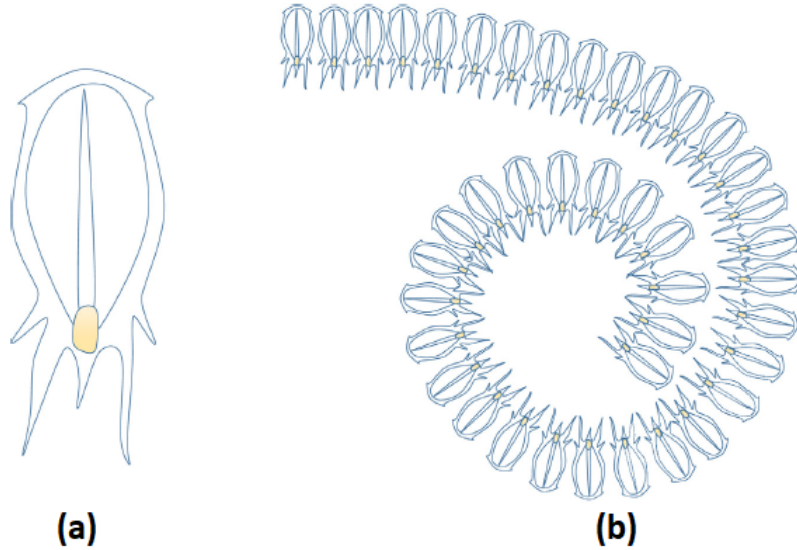


Figura 2.11: Salpa (a) e uma corrente de salpas (b) (MIRJALILI et al., 2017)

Por estar inserido em um ambiente de difícil acesso, os Salpas ainda não foram totalmente desvendados, mas algumas pesquisas revelaram um hábito curioso desses animais. Eles possuem um comportamento coletivo no qual se juntam em formato de correntes guiados por um líder (Fig. 2.11b). Os pesquisadores acreditam que a formação dessas correntes facilitam na locomoção dos Salpas.

(b) **Modelo matemático:**

O comportamento coletivo dos Salpas inspirou a criação da meta-heurística SSA por Mirjalili et al. (2017). SSA divide a população de Salpas em dois grupos: líder da corrente e seus seguidores. O líder é o agente de busca com a melhor aptidão e atrai os demais agentes para si. A posição do líder é atualizada conforme a equação (2.50).

$$X_{1,j}^{t+1} = \begin{cases} X_{p_j}^t + c_1((ub_j + lb_j)c_2 + lb_j) & \text{se } c_3 \geq 0,5 \\ X_{p_j}^t - c_1((ub_j + lb_j)c_2 + lb_j) & \text{se } c_3 < 0,5 \end{cases} \quad (2.50)$$

onde $X_{1,j}$ é a posição do líder na dimensão j , X_{p_j} é a posição da fonte de comida na dimensão j , c_2 e c_3 são coeficientes de valores aleatórios uniformes no intervalo de 0 a 1 e definem a direção pela qual o líder irá seguir. O coeficiente c_1 é responsável pelo equilíbrio entre a fase de exploração e de refinamento, por isso é o parâmetro mais importante do algoritmo. Seu valor é obtido pela equação (2.51).

$$c_1 = 2e^{-\left(\frac{4t}{Max_{iter}}\right)^2} \quad (2.51)$$

Os agentes seguidores atualizam suas posições de acordo com a lei de movimento de Newton:

$$X_{i,j} = \frac{1}{2}a\Delta t^2 + v_0\Delta t \quad (2.52)$$

onde $i \geq 2$, $X_{i,j}$ é a posição do i -ésimo seguidor na dimensão j , Δt é o tempo, v_0 é a velocidade inicial e a é a aceleração obtida por $a = \frac{v-v_0}{\Delta t}$ e v é obtido por $v = \frac{X_{i,j}^t - X_{i-1,j}^t}{\Delta t}$.

Considerando $v_0 = 0$ e que a cada iteração $\Delta t = 1$, a equação (2.52) se torna:

$$X_{i,j}^{t+1} = \frac{1}{2}(X_{i,j}^t - X_{i-1,j}^t) \quad (2.53)$$

Pode-se agora sintetizar a meta-heurística SSA no item a seguir.

(c) **Pseudo código:**

O pseudo código do algoritmo SSA pode ser visualizado pelo **Algoritmo 10**.

Algoritmo 10 : SSA

Entrada: S , Max_{iter} , lb e ub

Saída: posição do melhor agente de busca X_p e sua aptidão $F(X_p)$

- 1: **Início**
 - 2: Inicializa a população de salpas X_i ($i = 1, 2, \dots, S$)
 - 3: **repete**
 - 4: Calcula a aptidão de cada agente de busca
 - 5: $X_p =$ o melhor agente de busca
 - 6: Atualiza c_1 pela Eq. (2.51)
 - 7: **para** cada salpa (X_i) **faça**
 - 8: **se** ($i == 1$) **então**
 - 9: Atualiza a posição do líder pela Eq. (2.50)
 - 10: **senão**
 - 11: Atualiza a posição dos seguidores pela Eq. (2.53)
 - 12: **fim se**
 - 13: **fim para**
 - 14: **até** $t \geq Max_{iter}$
 - 15: **fim**
-

Na seção a seguir será descrita a teoria acerca da análise estatística utilizada neste trabalho para avaliar o desempenho das meta-heurísticas.

2.2 Análise Estatística de Desempenho

2.2.1 Princípios da estatística

Segundo Carvalho (2015), estatística é um ramo da matemática que estabelece métodos de coleta, organização, descrição e interpretação de um ou mais conjuntos de dados. Esses dados são obtidos pelo processo de amostragem, o qual a partir de uma população (conjunto completo) são retiradas amostras. Uma unidade de amostra é dita como uma observação do todo, de tal forma que carrega em si características da população de origem. Por conseguinte, pode-se extrair generalizações, cujo papel é da estatística inferencial.

De acordo com CORREIA (2003), estatística inferencial é o conjunto de técnicas, com certo grau de confiabilidade, a fim de obter informações sobre uma população a partir da análise de sua amostra. As características são obtidas através de estimadores dos parâmetros populacionais, extraídos da amostra (Fig. 2.12).

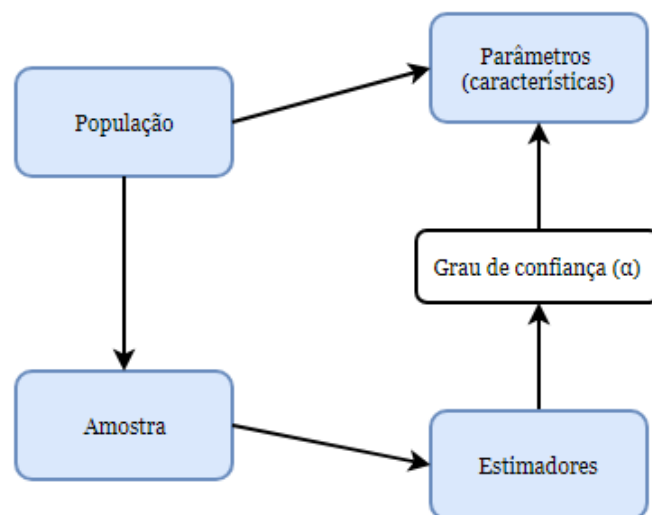


Figura 2.12: Inferência estatística da população

Não é possível determinar, com 100% de certeza, que os estimadores oriundos da amostra refletem as características da população, assim, determina-se que as informações possuem um índice de significância α o que garante $(1 - \alpha)\%$ de garantia sobre as conclusões apresentadas.

O dado amostral é denominado *variável aleatória*, pois resulta em um valor numérico a cada experimento. Ao analisar estatisticamente os dados, três parâmetros principais da população podem ser analisados, a média (μ), a variância (σ^2) e o desvio padrão (σ), e a partir da amostra observamos seus estimadores, respectivamente como \bar{x} , s^2 e s . Supondo que a amostra constitua n dados, o cálculo dos estimadores segue as equações (2.54) e (2.55),

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (2.54)$$

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1} \quad (2.55)$$

Outra forma de caracterizar uma amostra de dados é verificando a frequência dos resultados através de uma distribuição de probabilidades. Essa distribuição é conhecida como histograma e pode ser observada na Fig. 2.13.

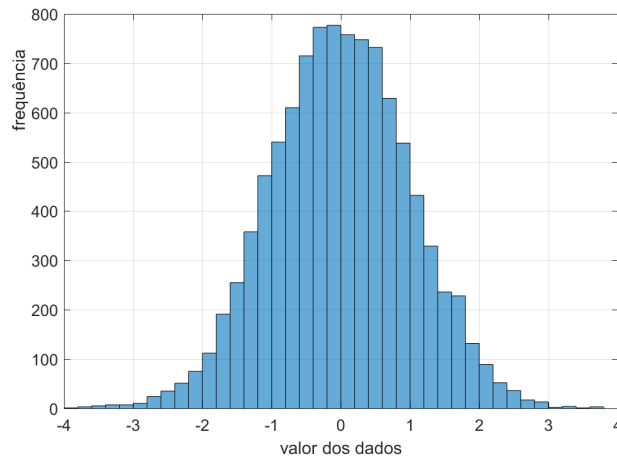


Figura 2.13: Histograma de um conjunto de 10.000 dados gerados pela função $randn(\cdot)$ do MatLab

Ademais é possível determinar uma função que associa a cada valor assumido pela variável aleatória à probabilidade do valor ocorrer, denominada função *distribuição acumulada* (Fig. 2.14).

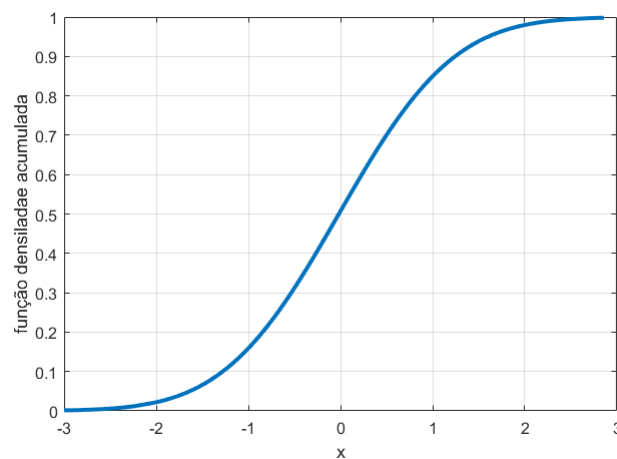


Figura 2.14: Função distribuição acumulada ($\mu = 0$ e $\sigma = 1$)

Para verificar a probabilidade de um valor da amostra X_a ser menor que x , temos $F(x) = P(X_a \leq x)$ e em caso de variável aleatória contínua a probabilidade é:

$$F(x) = \int_{-\infty}^x f(x)dx \quad (2.56)$$

Do histograma pode-se determinar uma curva que caracteriza a distribuição da amostra. Existem diversas distribuições de probabilidade que caracterizam processos físicos e outros eventos de amostragem, tais como distribuição *normal*, *Qui-quadrado* e *F de Snedecor*.

• **Distribuição normal:**

Para Soong (1986), a distribuição normal ou *Gaussiana* é a mais importante das distribuições, tanto na teoria quanto nas aplicações práticas. Vale saber então que uma variável aleatória é dita normal se sua densidade de probabilidade possui a forma da equação (2.57).

$$f_{X_a}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{(x - \mu)^2}{2\sigma^2} \right], \quad -\infty < x < \infty \quad (2.57)$$

A Fig. 2.15 mostra a função de distribuição de probabilidade (*fdp*) na curva em formato de sino, simétrica em relação ao eixo das ordenadas, com valores de μ e σ distintos (SOONG, 1986).

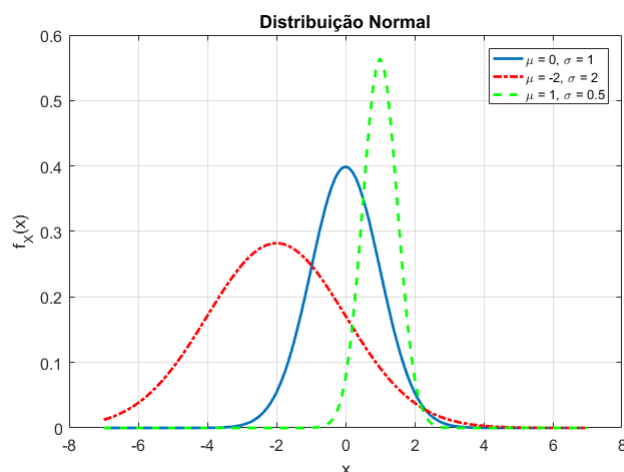


Figura 2.15: Distribuição normal

Denota-se $N(\mu, \sigma^2)$ uma distribuição normal com média μ e variância σ^2 . Uma variável aleatória normal com $\mu = 0$ e $\sigma = 1$, como em uma das curvas na Fig. 2.15, é chamada de variável aleatória normal padrão e denominada Z . Na subseção 2.2.3 será mostrado como transformar uma variável normal em uma com distribuição normal padrão.

• **Qui-quadrado:**

Uma variável aleatória X_a tem distribuição qui-quadrado com n graus de liberdade² (n é igual ao número total de dados menos uma unidade) quando sua *fdp* é obtida por:

$$f_{X_a}(x) = \frac{1}{2^{\frac{n}{2}} \Gamma(n/2)} \cdot x^{\frac{n}{2}-1} \exp\left(-\frac{x}{2}\right), \quad n > 0, x > 0 \quad (2.58)$$

onde $\Gamma(\omega) = \int_0^\infty x^{\omega-1} e^{-x} dx, \omega > 0$. Para uma variável aleatória X_a com distribuição qui-quadrada

²Grau de liberdade é o número de valores do conjunto de dados que pode variar após terem sido impostas certas restrições, como média ou variância

denota-se $X_a \sim \chi_n^2$. Ainda, a distribuição qui-quadrada pode ser definida como a soma de variáveis normais padronizadas ao quadrado, expresso como:

$$\chi_n^2 = \sum_{i=1}^n X_{a_i}^2, \text{ com } X_{a_i} \sim N(0, 1) \quad (2.59)$$

Na Fig. 2.16 pode ser observada *fdp* da equação (2.58) com diversos graus de liberdade (SANDERS, 2009).

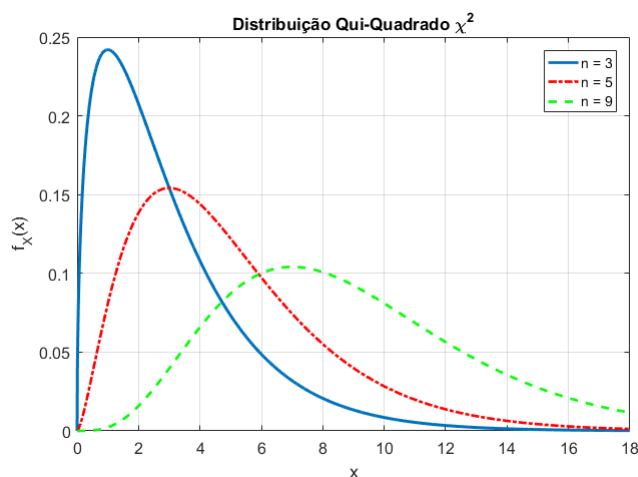


Figura 2.16: Distribuição Qui-Quadrado χ^2

• **Distribuição F de Snedecor:**

Uma variável aleatória X_a tem distribuição F de Snedecor com n graus de liberdade no numerador e m no denominador quando sua *fdp* é obtida por:

$$f_{X_a}(x) = \frac{\Gamma(\frac{m+n}{2}) (\frac{m}{n})^{\frac{m}{2}} \cdot x^{\frac{m}{2}-1}}{\Gamma(\frac{m}{2}) \Gamma(\frac{n}{2}) (\frac{m}{n}x + 1)^{\frac{m+n}{2}}}, \quad x > 0 \quad (2.60)$$

Denota-se como $X_a \sim F(m, n)$ e o gráfico que representa a *fdp* F , para alguns valores de m e n , pode ser visto na Fig 2.17 (JOHNSON; KEMP; KOTZ, 2005).

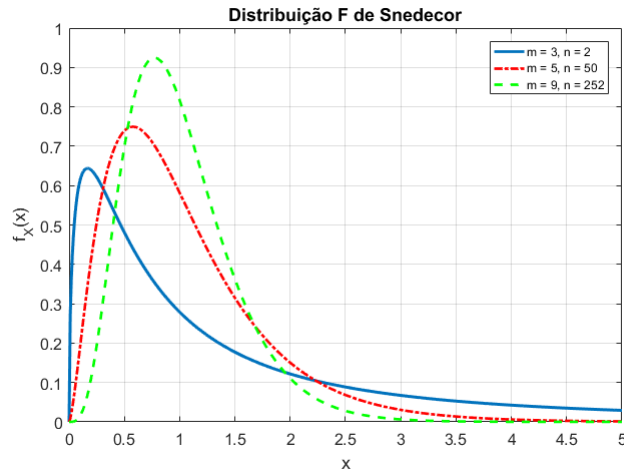


Figura 2.17: Distribuição F de Snedecor

2.2.2 Teste de hipótese

Devore (2010) diz que um teste de hipótese, ou hipótese estatística, é uma afirmação acerca do valor de um parâmetro, dos valores de vários parâmetros ou até a identificação da distribuição de probabilidade da amostra. Independentemente do problema, o teste de hipótese é baseado em duas suspeições, uma contrária a outra. Pode-se definir como hipótese que uma certa população possui $\mu = 0,5$, assim a outra seria $\mu \neq 0,5$, para análise de apenas uma única distribuição, ou dada duas populações μ_1 e μ_2 , descobrir se possuem distribuições iguais ou diferentes. O teste tem como objetivo verificar qual das hipóteses é verdadeira. Inicialmente uma delas é favorecida, alegando veracidade e tal alegação será rejeitada caso haja evidência que a contradiga.

Defini-se a hipótese nula (H_0) como a alegação assumida verdadeira e hipótese alternativa (H_1) como a *contradição* de H_0 . A exemplo do parágrafo anterior, um teste de hipótese pode ser como $H_0 : \mu = 0,5$ e $H_1 : \mu \neq 0,5$. Apesar de estar bem definida, a hipótese nula não é rejeitada por qualquer valor diferente de 0,5, pode-se aceitar que μ esteja em certo intervalo próximo, como por exemplo $0,3 \leq \mu \leq 0,7$, conhecido como intervalo de confiança, assim, definindo a região de rejeição para $\mu < 0,3$ e $\mu > 0,7$. Essa região está relacionada ao nível de significância (α) que o teste deseja aplicar.

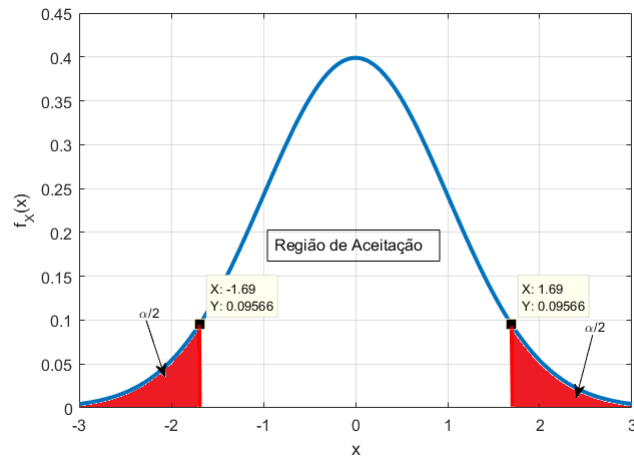


Figura 2.18: Intervalo de confiança para $\alpha = 0.5$

Mesmo com um intervalo de confiança escolhido de forma sensata, a característica da amostra pode levar a afirmações errôneas acerca da população. Dois erros importantes no estudo dos testes de hipótese podem ocorrer: **erro tipo I**, ocorre na rejeição da hipótese nula H_0 quando ela é verdadeira; **erro tipo II**, ocorre na não rejeição de H_0 quando falsa. O grau de confiança de um teste é a probabilidade de ocorrer o erro do tipo I, ou seja, quanto menor for α menor é a chance de definir H_0 falsa quando verdadeira. O complemento de α é o β que é a probabilidade da ocorrência do erro tipo II. A relação entre eles é $\alpha = 1 - \beta$. A síntese dos tipos de erros pode ser vista na Tab. 2.1.

Tabela 2.1: Ocorrência de erros no teste de hipótese

	Realidade	
	H_0 verdadeira	H_0 falsa
Aceitar H_0	Resultado correto	Erro tipo II (β)
Rejeitar H_0	Erro tipo I (α)	Resultado correto

Para eliminar completamente a ocorrência desses erros é necessário que o teste seja feito com o conjunto total da população, o que é impraticável em determinados casos. Porém, a escolha correta do tamanho da amostra diminui a chance dos mesmos acontecerem.

O resultado de um teste de hipótese pode ser obtido, então, com a aceitação ou não de H_0 dado um nível de significância α , porém de acordo com Montgomery e Runger (2010) esse tipo de conclusão é inadequada porque não diz ao pesquisador se o valor calculado do teste estatístico está perto do intervalo de confiança ou se extrapolou a região de rejeição. O autor ressalta ainda que não é possível medir o risco ao aplicar somente um α fixo.

Para resolver esse problema, criou-se a abordagem do **valor-p** (ARSHAM, 1988). O valor-p é uma medida de quanta evidência você tem *contra* a hipótese nula. Quanto menor o *valor-p*, mais evidência terá de que H_0 é falsa. Diferentemente do α , o *valor-p* é extraído diretamente da amostra, mas, a interpretação de um teste de hipótese está diretamente ligado aos valores do primeiro, como mostrado na Tab. 2.2. O

cálculo do seu valor varia de acordo com o teste e distribuição adotados, os quais serão apresentados nas subseções a seguir.

Tabela 2.2: Significado do *valor-p*

<i>Valor-p</i>	Interpretação	Decisão
$p \leq \alpha$	Forte evidência contra H_0	Rejeita H_0
$p > \alpha$	Pouca ou nenhuma evidência contra H_0	Aceita H_0

2.2.3 Testes paramétricos e não-paramétricos

Existem, na literatura, diferentes testes para vários tipos de problemas. Há o teste que é realizado para confirmar a média de uma distribuição normal quando a variância (σ^2) da população é conhecida, usando a distribuição normal padrão:

$$Z_0 = \frac{\bar{X}_a - \mu_0}{\sigma/\sqrt{n}} \quad (2.61)$$

onde n é o número de elementos da amostra. Quando a variância da população não é conhecida, o teste t é realizado utilizando a variância da amostra s^2 :

$$T_0 = \frac{\bar{X}_a - \mu_0}{s/\sqrt{n}} \quad (2.62)$$

Os procedimentos acima podem ser usados também para testes acerca de duas distribuições normais (X_1 e X_2), com populações de variância conhecida (Eq. (2.63)) e variância desconhecida (Eq. (2.64)).

$$Z_0 = \frac{\bar{X}_1 - \bar{X}_2 - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \quad (2.63)$$

$$T_0 = \frac{\bar{X}_1 - \bar{X}_2 - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (2.64)$$

Os testes mostrados pelas equações de (2.61) a (2.64) são aplicados quando as amostras em questão apresentam distribuição normal. Tais procedimentos são denominados de **testes estatísticos paramétricos**. Para definir se uma amostra possui distribuição normal outros procedimentos são realizados e serão mostrados na subseção 2.2.4. Além da exigência de normalidade na distribuição, de acordo com García et al. (2009) há mais dois requisitos para o uso dos testes paramétricos acerca de múltiplas amostras: *independência dos dados e homogeneidade da variância*.

A independência dos dados indica a influência que a captação de uma amostra aplica sobre a outra. Neste trabalho, as meta-heurísticas são testadas em instantes distintos e parâmetros autônomos, ou seja, todos os dados apresentados são independentes entre si.

A homogeneidade da variância requer que as amostras de uma mesma população devem possuir variância iguais. Os testes para essa verificação serão abordados na subseção 2.2.5.

Casos em que as amostras não apresentem distribuições normais, independência e homogeneidade de variância, outros procedimentos de teste podem ser usados, conhecidos como **testes estatísticos não-paramétricos** ou *livres de distribuição* (MONTGOMERY; RUNGER, 2010). Siegel e Jr (1975) cita algumas vantagens do uso de estatística não-paramétrica em vez da paramétrica:

1. Se a distribuição da população não é totalmente conhecida os métodos não-paramétricos podem ser usados com amostras de pequeno tamanho, com um alto grau de eficiência;
2. Os testes não-paramétricos podem ser mais relevantes em pesquisas particulares por fazer menos suposições acerca dos dados;
3. A estatística não-paramétrica pode ser usada em aplicações categóricas de dados, ou seja, medidas nominais;
4. Há testes não-paramétricos que tratam amostras obtidas de populações distintas. Os paramétricos, em geral, não tratam tais dados sem suposições não reais;
5. Métodos não-paramétricos são mais fáceis de entender do que os paramétricos, além de os resultados daqueles serem mais diretamente interpretados que estes.

Sem acesso à população que originou as amostras, não é possível afirmar que ela tem exatamente distribuição normal ou qualquer outra, apenas é dito que a amostra é aproximadamente distribuída de tal forma (normal ou não). Os testes que indicam normalidade da distribuição dos dados são abordados na subseção a seguir. Além disso, o teste que indica a homogeneidade da variância de uma amostra será mostrado na subseção 2.2.5. Como veremos mais adiante, nem todas as populações possuem distribuição normal e nem todas garantem a homogeneidade da variância. Assim, nestes casos o uso dos testes não-paramétricos são recomendados, os métodos utilizados neste trabalho serão detalhados na subseção 2.2.6.

2.2.4 Teste de normalidade

(a) Hipóteses

Como o próprio nome diz, teste de normalidade indica se a população (da qual a amostra é proveniente) possui distribuição normal ou não:

- $H_0 : X_a \in N(\mu, \sigma)$ e $H_1 : X_a \notin N(\mu, \sigma)$.

Neste trabalho foram utilizados três testes de normalidade com abordagens distintas, os quais são descritos a seguir.

(b) Descrição dos testes

(i) Função de distribuição empírica

Uma maneira de analisar a distribuição de uma amostra é compará-la com uma *fdp* acumulada conhecida. Para isso, calcula-se a **função de distribuição empírica** da amostra, primeiro ordenando (y_1, y_2, \dots, y_n) os valores da amostra (x_1, x_2, \dots, x_n) . Considerando i a frequência de y_k , a função de distribuição empírica é calculada como (CONOVER, 1999):

$$FDE_n(y) = \begin{cases} 0, & y < y_1 \\ i/n, & y_k \leq y < y_{k+1}, \quad k = 1, 2, \dots, n-1 \\ 1, & y \geq y_n \end{cases} \quad (2.65)$$

A função $FDE_n(x)$ é sempre composta por degraus e cabe aos testes, baseados nela, verificar a similaridade com a *fdp* acumulada conhecida $FDE_0(x)$ (Fig. 2.19).

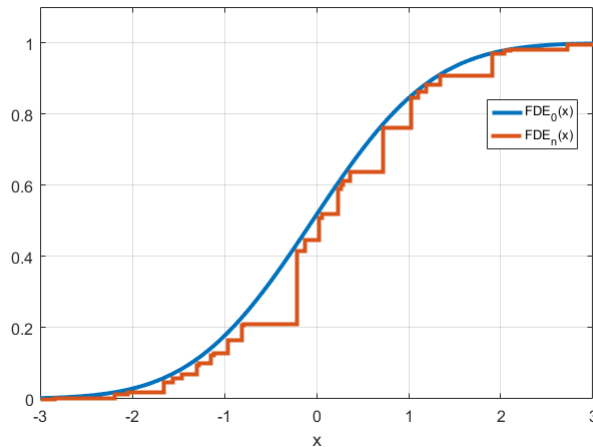


Figura 2.19: Distribuição conhecida $FDE_0(x)$, distribuição empírica $FDE_n(x)$

O teste **Kolmogorov-Smirnov Lilliefors** (KS-L) é baseado na FDE, como descrito no item a seguir.

(ii) **Kolmogorov-Smirnov Lilliefors** (KS-L)

KS-L é uma extensão do teste **Kolmogorov-Smirnov** (KS). KS é um método clássico baseado no conceito da FDE (DŽOLMOGOROV, 1933). Nele calcula-se a maior distância entre $FDE_n(x)$ com $FDE_0(x)$ fazendo:

$$D_n = \max\{|D_n^-|, |D_n^+|\} \quad (2.66)$$

do qual $D_n^- = FDE_n(x_{k-1}) - FDE_0(x_k)$ e $D_n^+ = FDE_n(x_k) - FDE_0(x_k)$. O valor da estatística KSz é então calculado:

$$KSz = \sqrt{n}D_n \quad (2.67)$$

A hipótese nula H_0 é rejeitada se $KSz < K_\alpha$, em um nível α de significância. K_α é encontrado pela distribuição de Kolmogorov e pode ser vista em (SMIRNOV, 1948). A realização do teste KS segue os passos:

Passo 01: vetor da amostra \mathbf{x}

Passo 02: ordena \mathbf{x} , $y_i = \text{sort}(\mathbf{x})$

Passo 03: $i=1,2,\dots,n$

Passo 04: $FDE_n(y) = i/n$

Passo 05: $\bar{y} = \sum y_i/n$

Passo 06: $\sigma(y) = \sqrt{\sum (y_i - \bar{y})^2/n - 1}$

Passo 07: $z_i = (y_i - \bar{y})/\sigma(y)$

Passo 08: $FDE_0(y) = u_i = \phi(z_i)$

Passo 09: $D^+ = \text{abs}(FDE_0(y) - i/n)$

Passo 10: $D^- = \text{abs}(FDE_0(y) - (i - 1)/n)$

Passo 11: $D_n = \max(D^+, D^-)$

Passo 12: $KSz = \sqrt{n}D_n$

O teste KS-L provém de uma correção do teste KS, para quando os parâmetros de média e variância são desconhecidos, apresentado por Lilliefors (1967). O *valor-p* é calculado por um aproximação numérica elaborada por Dallal e Wilkinson (1986). Os passos, após aqueles do teste KS, para o cálculo do *valor-p* são:

Passo 13: encontrar os valores críticos de D_{20} e D_{15} que correspondem a n (amostras) da tabela formulada por (DALLAL; WILKINSON, 1986). Se n está entre dois valores (n_{sup} e n_{inf}), os valores de D_{20} e D_{15} são encontrados pela seguinte interpolação linear:

$$D_{20} = D_{20_{inf}} + (n + n_{inf}) \frac{D_{20_{sup}} - D_{20_{inf}}}{n_{sup} - n_{inf}} \quad (2.68)$$

$$D_{15} = D_{15_{inf}} + (n + n_{inf}) \frac{D_{15_{sup}} - D_{15_{inf}}}{n_{sup} - n_{inf}} \quad (2.69)$$

Passo 14: Calcular o valor de a_1, b_1, c_1, a_2, b_2 e c_2 :

$$\begin{aligned} a_1 &= -7,01256(n + 2,78019) \\ b_1 &= 2,99587\sqrt{n + 2,78019} \\ c_1 &= 2,1804661 + 0,974598/\sqrt{n} + 1,67997/n \\ a_2 &= -7,90289126054n^{0,98} \\ b_2 &= 3,180370175721n^{0,49} \\ c_2 &= 2,2947256 \end{aligned} \quad (2.70)$$

Passo 15: Calcular o valor crítico D_{10} :

$$D_{10} = \begin{cases} \frac{-b_1 - \sqrt{b_1^2 - 4a_1c_1}}{2a_1} & e \ a = a_1, \ b = b_1, \ c = c_1 \ se \ n \leq 100 \\ \frac{-b_2 - \sqrt{b_2^2 - 4a_2c_2}}{2a_2} & e \ a = a_2, \ b = b_2, \ c = c_2 \ se \ n > 100 \end{cases} \quad (2.71)$$

$$Passo 16: \text{valor-}p = \begin{cases} 0, 10 & \text{se } D_n = D_{10} \\ \exp(aD_n^2 + bD_n + c - 2, 3025851) & \text{se } D_n > D_{10} \\ 0, 15 + (D_n - D_{15})[(0, 10 - 0, 15)/(D_{10} - D_{15})] & \text{se } D_n \geq D_{15} \\ 0, 20 + (D_n - D_{20})[(0, 15 - 0, 20)/(D_{15} - D_{20})] & \text{se } D_n \geq D_{20} \\ > 0, 20 & \text{se } D_n \leq D_{10} \end{cases}$$

(iii) **Teste Shapiro-Wilk (SW)**

O teste SW é baseado em regressão e correlação dos dados (SHAPIRO; WILK, 1965). A estatística do teste SW é calculada como:

$$W_s = \frac{(\sum_{i=1}^n a_i y_i)^2}{\sum_{i=1}^n (y_i - \bar{x})^2} \quad (2.72)$$

onde y_i é o i -ésimo dado do conjunto x_i ordenado e a_i é o i -ésimo elemento do vetor:

$$\mathbf{a}' = \frac{\mathbf{m}'\mathbf{V}^{-1}}{(\mathbf{m}'\mathbf{V}^{-1}\mathbf{V}^{-1}\mathbf{m})^{1/2}} \quad (2.73)$$

onde $\mathbf{m}' = (m_1, m_2, \dots, m_n)$ é o vetor de valores esperados para a estatística normal e $\mathbf{V} = [cov(y_i, y_j)]$ é a matriz de covariância estatística. O cálculo de \mathbf{m}' pode ser encontrado em (ÖNER; KOCAKOÇ, 2017).

Os passos que calculam o *valor-p* do teste SW é dado como:

Passo 01: calcula o vetor \mathbf{a}' pela equação (2.73)

Passo 02: calcula a estatística W_s pela equação (2.72)

Passo 03: calcula a normalização de W_s (ROYSTON, 1982)

$$Z = \frac{\ln(1-W_s) - \mu}{\sigma},$$

onde:

$$\mu = -1, 5861 - 0, 31082\ln(n) - 0, 083751\ln^2(n) + 0, 0038915\ln^3(n) \text{ e}$$

$$\sigma = \exp(-0, 4803 - 0, 082676\ln(n) + 0, 0030302\ln^2(n))$$

Passo 04: $\text{valor-}p = P(Z \geq z) = 1 - \phi(|z|)$

(iv) **Teste D'Agostinho e Pearson (DAP)**

O teste D'Agostinho e Pearson é baseado nos momentos da amostra $(\sqrt{b_1}, b_2)$ (D'AGOSTINO; PEARSON, 1973). O teste estatístico DAP é calculado pela expressão:

$$DAP = Z^2(\sqrt{b_1}) + Z^2(b_2) \quad (2.74)$$

Os testes $Z(\sqrt{b_1})$ e $Z(b_2)$ são próximos à distribuição normal e seu cálculo pode ser verificado em (ÖNER; KOCAKOÇ, 2017). Dessa forma, os passos para o cálculo do *valor-p* do teste DAP são:

Passo 01: calcula as estatísticas $Z(\sqrt{b_1})$ e $Z(b_2)$

Passo 02: calcula a estatística DAP pela equação (2.74)

Passo 03: calcula o *valor-p* de uma distribuição qui-quadrada com 2 graus de liberdade.

(c) **Interpretação**

Para todos os testes de normalidades analisados anteriormente se $\text{valor-p} > \alpha$ então H_0 é aceita como verdadeira, indicando a normalidade da população em estudo. Todos esses testes foram realizados em código MatLab que pode ser obtido diretamente do Apêndice no trabalho de Öner e Kocakoç (2017).

2.2.5 Teste de homogeneidade de variância

(a) **Hipótese**

Segundo García et al. (2009) e RIBOLDI et al. (2014), o método de Levene possui maior eficácia no teste de homogeneidade de variância quando aplicado a dados que não apresentam distribuição normal.

Esse teste proposto por Levene (1961) possui como hipótese nula (H_0) que determinado grupo de amostras possuem mesma variância. A hipótese alternativa H_1 indica que no mínimo uma amostra do grupo possui variância diferente das demais.

(b) **Descrição do teste**

O teste de Levene começa com o cálculo de W na equação (2.75).

$$W = \frac{(N - k)}{(k - 1)} \cdot \frac{\sum_{i=1}^k N_i (Z_{i,\cdot} - Z_{\cdot\cdot})^2}{\sum_{i=1}^k \sum_{j=1}^{N_i} (Z_{i,j} - Z_{i,\cdot})^2} \quad (2.75)$$

onde N_i é o número de amostras no i -ésimo grupo, N é a soma das amostras de todos os grupos, k é a quantidade de grupos, $Z_{i,j}$ é o módulo da diferença entre a variável aleatória da amostra j e grupo i ($Y_{i,j}$) com a média do grupo i , ou seja, $Z_{i,j} = |Y_{i,j} - \bar{Y}_i|$. $Z_{i,\cdot}$ é a média de $Z_{i,j}$ para o grupo i , $Z_{\cdot\cdot}$ é a média de todos os $Z_{i,j}$. A expressão de ambas são mostradas nas equações (2.76) e (2.77).

$$Z_{i,\cdot} = \frac{1}{N_i} \sum_{j=1}^{N_i} Z_{i,j} \quad (2.76)$$

$$Z_{\cdot\cdot} = \frac{1}{N} \sum_{i=1}^k \sum_{j=1}^{N_i} Z_{i,j} \quad (2.77)$$

O teste estatístico W é F-distribuído, o *valor-p* é encontrado pela distribuição acumulada F com $k - 1$ graus de liberdade no numerador e $N - k$ no denominador.

(c) **Interpretação**

Neste trabalho foi utilizada a função $fcd(f(W, k-1, N-k, 'upper'))$ do MatLab, a qual retorna o *valor-p* do teste. Se $\text{valor-p} > \alpha$ indica que as amostras possuem mesma variância, ou seja, aceitam H_0 (ÖNER; KOCAKOÇ, 2017).

O teste de homogeneidade de variância de Levene compara apenas a ocorrência de diferenças na variância e é usado como requisito para o uso de métodos paramétricos. Na subseção a seguir será abordado o teste não-paramétrico para múltiplas amostras ($n > 2$) e busca indicar diferenças significativas na média das amostras.

2.2.6 Teste não-paramétrico do desempenho

Em se tratando da análise de meta-heurísticas, frequentemente deseja-se comparar o desempenho deles em diversos problemas. Esse tipo de teste avalia a média de cada amostra (meta-heurística) do grupo para um número adequado de problemas. Para a minimização de problemas a amostra com a menor média possui o melhor desempenho. No grupo dos testes paramétricos, o mais usado para este propósito é o ANOVA (FISHER, 1956). Alternativamente, o método não-paramétrico adequado para a realização dessas comparações é o teste proposto por Friedman (1937) mostrado no item a seguir.

(a) Teste de Friedman

(i) Hipótese

O teste de Friedman é um método não-paramétrico análogo ao ANOVA e seu objetivo é dizer, em um conjunto de n amostras ($n > 2$), se existe pelo menos duas que representem populações de médias distintas. Dessa forma, é possível detectar diferenças significativas entre o comportamento de duas ou mais meta-heurísticas.

A hipótese nula H_0 indica que as populações de todas as amostras possuem médias iguais, afirmando a familiaridade delas. A hipótese alternativa simplesmente nega a hipótese nula, ou seja, evidencia a diferença entre as médias das populações, indicando que pelo menos uma das amostras possui média diferente.

(ii) Descrição do teste

O teste de Friedman é baseado no ranqueamento dos dados, realizando os seguintes passos:

- i. Organiza os dados obtidos em pares de cada meta-heurística para cada problema;
- ii. Para cada problema i , realiza o ranqueamento dos valores de 1 (melhor resultado) a k (pior resultado), onde k é o número de meta-heurísticas. O ranking é denominado como $r_{i,j}$ com $1 \leq j \leq k$;
- iii. Para cada meta-heurística j , calcular a média dos rankings obtidos em todos os problemas e obter o valor final $R_j = \frac{1}{n} \sum_{i=1}^n r_{i,j}$.

Logo, o teste de Friedman atribui o menor valor do ranking à meta-heurística com melhor desempenho. Para verificar as hipóteses de teste, a estatística de Friedman é calculada como:

$$\chi_f^2 = \frac{12n}{k(k+1)} \left[\sum_{j=1}^k R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (2.78)$$

o qual é distribuído de acordo com a distribuição χ^2 com $k - 1$ graus de liberdade, para n e k grandes o suficiente $n > 10$ e $k > 5$ (DEMŠAR, 2006).

Porém, Iman e Davenport (1980) mostrou que o teste estatístico de Friedman é muito conservativo e abordou um novo teste:

$$F_f = \frac{(n-1)\chi_f^2}{n(k-1) - \chi_f^2} \quad (2.79)$$

o qual é F distribuído com $k - 1$ e $(k - 1)(n - 1)$ graus de liberdade. O *valor-p* desses testes são obtidos das tabelas de suas distribuições em conjunto com os graus de liberdade.

(iii) Interpretação

Para as duas estatísticas apresentadas, quando o *valor-p* $\leq \alpha$ a hipótese nula H_0 é rejeitada, determinando que ao menos uma das meta-heurísticas possui média diferente e não tem familiaridade com os demais.

(b) Comparação 1x1 das meta-heurísticas

Além do testes das equações (2.78) e (2.79), necessita-se encontrar a diferença entre pares das amostras em estudo, para isso outro teste pode ser realizado com o ranking de Friedman. Nele pode-se comparar duas meta-heurísticas em específico (i e j) e começa pelo cálculo de:

$$z = (R_i - R_j) / \sqrt{\frac{k(k+1)}{6n}} \quad (2.80)$$

onde z é o valor utilizado para encontrar o *valor-p* da tabela de distribuição normal, que posteriormente é comparado com o α escolhido, rejeitando a hipótese nula caso *valor-p* $\leq \alpha$. Assim, dado $R_i < R_j$ e *valor-p* $< \alpha$ o teste indica que a meta-heurística i teve melhor desempenho que j , já que está em melhor posição no *ranking* de Friedman e não possuem familiaridade (médias distintas).

O *valor-p* obtido pelo valor de z da equação (2.80) reflete a comparação entre as meta-heurísticas i e j , sem levar em conta a comparação remanescente de todo o conjunto em estudo. Por conta disso esse resultado é denominado **valor-p não ajustado**

(c) Erro acumulativo do valor-p não ajustado

Considerando um grupo de k meta-heurísticas sendo comparadas e um nível de significância α , como mostrado na tabela 2.1, a probabilidade de não ocorrer um erro do tipo I é $(1 - \alpha)$. Já a probabilidade de não ocorrer esse mesmo erro em $k - 1$ comparações (meta-heurística i com todos os outros) é $(1 - \alpha)^{(k-1)}$.

Dessa forma, a ocorrência de um ou mais erros tipo I terá a probabilidade de $1 - (1 - \alpha)^{(k-1)}$. Para $\alpha = 0,05$ e $k = 10$, esse cálculo é de aproximadamente 0,37, valor consideravelmente alto. Essa probabilidade de erros acumulados em múltiplas comparações é denominada FWER (*Family-Wise Error Rate* em inglês).

A fim de lidar com esse problema usa-se os *valores-p ajustados* (APVs - *adjusted p-values* em inglês). Os APVs são calculados pelos procedimentos post-hoc, métodos esses que levam em con-

sideração o controle da FWER do conjunto. Os procedimentos post-hoc abordados neste trabalho serão descritos na próxima subseção (DERRAC et al., 2011).

2.2.7 Procedimentos *post-hoc*

(a) Descrição dos procedimentos

A aplicação de um procedimento *post-hoc* tem como objetivo obter os APVs apropriados para realizar a comparação de múltiplas meta-heurísticas e determinar o grau de rejeição de cada hipótese testada. A hipótese nula H_0 afirma que as duas meta-heurísticas sendo analisadas são da mesma família e têm o mesmo desempenho, portanto, a hipótese alternativa afirma que as meta-heurísticas não são da mesma família. Derrac et al. (2011) realizaram um estudo dos testes não-paramétricos, apresentando o uso de alguns procedimentos post-hoc para comparações $N \times N$, dentre eles o **Nemenyi** (NEMENYI, 1963) e **Shaffer** (SHAFFER, 1986):

- (a) **Nemenyi**: este procedimento é classificado como um método *single step* (um único passo, em inglês) pois transforma o *valor-p* não ajustado em APV com apenas uma expressão, dividindo o primeiro pelo número de comparações apresentadas, $m = k(k - 1)/2$. O cálculo do APV da hipótese i é feito por:

$$APV_i = \min[v, 1], \text{ onde } v = m \cdot p_i \quad (2.81)$$

- (b) **Shaffer**: este procedimento é classificado como um método *step-down* pois analisa as hipóteses do menor valor-p ao maior. A rejeição da hipótese H_i ocorre se $p_i < \alpha/t_i$, onde t_i é o número máximo de hipóteses que pode ser verdadeira dado que alguma hipótese H_j ($j \leq i$) seja falsa. Os APVs do procedimento Shaffer são obtidos da seguinte forma:

$$APV_i = \min[v, 1], \text{ onde } v = \max[t_j p_j : 1 \leq j \leq i] \quad (2.82)$$

(b) Interpretação

Nesses procedimentos, se $APV_i \leq \alpha$ rejeita-se a hipótese nula H_0 , ou seja, as meta-heurísticas comparadas não são da mesma família, o que indica não terem o mesmo desempenho. Os testes e procedimentos *post-hoc* anunciados foram executados através de um *software* muito utilizado em testes estatísticos. Esse *software* é relatado na subseção a seguir.

2.2.8 Software KEEL

KEEL³ (*Knowledge Extraction based on Evolutionary Learning*) é um *software* livre desenvolvido em Java por um grupo de pesquisadores da Espanha e Reino Unido (ALCALÁ-FDEZ et al., 2009; ALCALÁ-FDEZ et al., 2011; TRIGUERO et al., 2017), que pode realizar diversos experimentos em mineração de dados, tais como regressão, classificação, aprendizagem não-supervisionada, entre outros. Importante

³<http://www.keel.es/>

mencionar que esse software é a primeira ferramenta do tipo que contém biblioteca de algoritmos evolucionários com código aberto. KEEL fornece uma interface gráfica simples (Fig. 2.20) a fim de realizar experimentos com diferentes conjuntos de dados provenientes de algoritmos (especialmente os evolucionários) e avaliar os seus comportamentos.



Figura 2.20: Tela inicial

Na aba de módulos (Fig. 2.21) existe a opção *Non-Parametric Statistical Analysis*, nela KEEL fornece um conjunto de procedimentos estatísticos para comparações múltiplas e em pares. Dentro do software vários métodos paramétricos e não-paramétricos podem ser utilizados. Os dados dos algoritmos a serem analisados podem ser carregados de outras ferramentas através de arquivos *excel* pela opção *Load Data* na Fig. 2.22.

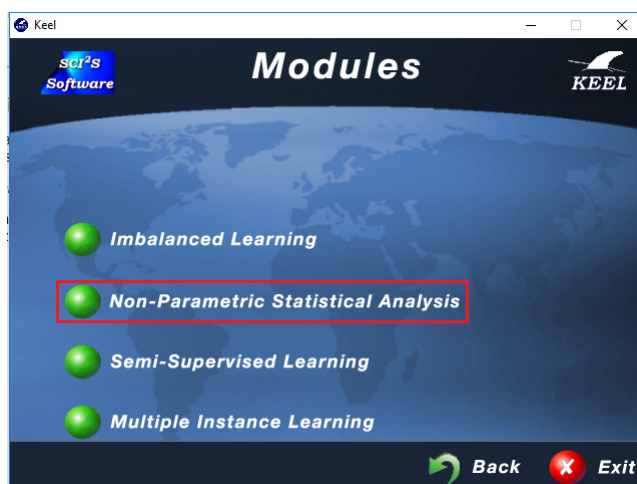


Figura 2.21: Tela de módulos

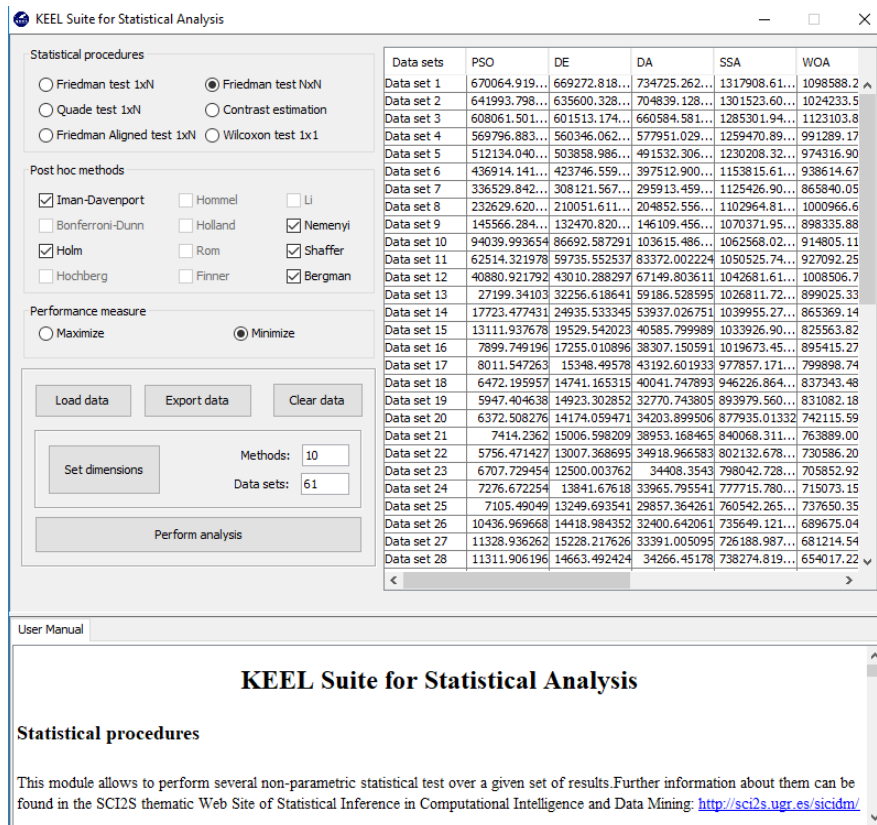


Figura 2.22: Tela dos testes estatísticos

Todos os testes estatísticos deste trabalho foram realizados nesta ferramenta.

2.3 Controle Preditivo baseado em Modelo (MPC)

O controle preditivo baseado em modelo (MPC) é uma estratégia de controle que usa do conhecimento do modelo dinâmico do sistema para prever seu comportamento ao longo de um conjunto de ações de controle, em um horizonte de predição (N) definido, e determinar a ação ótima (instante atual) que resulte na saída desejada. Criado por volta da década de setenta, o MPC foi proposto inicialmente para sistemas de entradas e saídas uni-variáveis (SISO, do inglês *Single Input Single Output*), contudo logo foi estendido para sistemas multi-variáveis (MIMO, do inglês *Multiple Input Multiple Output*), para maiores informações sobre essas abordagens consultar (WANG, 2009). Por essas características, o controle preditivo baseado em modelo tornou-se uma ferramenta muito aplicada em diversos problemas complexos, principalmente em sistemas não lineares (ALAMIR, 2013). Neste caso, o controle preditivo é denominado de NMPC ou *Non-Linear MPC*.

Outra importante característica do MPC é o uso sistemático de restrições no modelo do sistema. Essas podem ser atribuídas às características físicas de atuadores, restrições de segurança como variáveis de pressão e temperatura, contratuais (qualidade e dimensionamento do produto), restrições relacionadas à legislação (limite máximo de emissões), restrições de tempo, dentro outros. Por conta disso, muitos problemas de sistemas reais podem ser resolvidos com o MPC e na literatura é possível encontrá-los como

em controle aeroespacial (HARTLEY et al., 2012; HARTLEY et al., 2014; LING; WU; MACIEJOWSKI, 2008) e processos químicos (YAMASHITA; ZANIN; ODLOAK, 2016; ALI, 2001).

Ademais, o MPC é uma estratégia na qual a ação de controle é obtida de forma on-line, a cada instante de amostragem, o que difere do controle convencional, o qual a lei de controle é pré-calculada (RAWLINGS; MAYNE, 2009).

Neste trabalho, o controle preditivo é usado como ferramenta para dois dos três estudos de caso na avaliação dos algoritmos bioinspirados com modelos de sistemas não-lineares, diante disso, serão apresentados seus conceitos básicos para o entendimento de sua aplicação. Para uma compreensão mais profunda do assunto, sugere-se a leitura das seguintes referências (MACIEJOWSKI, 2002; RAWLINGS; MAYNE, 2009; ALAMIR, 2013).

Na próxima subseção será abordada a metodologia do controle preditivo.

2.3.1 Metodologia do MPC

A estratégia de controle do MPC, em síntese, segue quatro passos (indicados abaixo) aplicados a cada instante de decisão (ou período de amostragem), um desses instantes de decisão é ilustrado na Fig. 2.24

- 1: **para** instante de decisão k **faça**
- 2: Verificar a(s) variável(is) de estado do sistema;
- 3: Calcular a sequência de ações futuras que minimizem a função custo do sistema;
- 4: Aplicar a primeira ação de controle da sequência ótima no intervalo do instante $[k, k+1]$;
- 5: Passar para o instante $k=k+1$;
- 6: **fim para**

Outra forma de verificar a metodologia do MPC é através de um diagrama de blocos, conforme mostrado na Fig. 2.23. Nela pode-se verificar que o modelo do sistema é utilizado para prever as saídas, mediante os estados atuais (entradas e saídas anteriores) e as entradas futuras (ação de controle ótima).

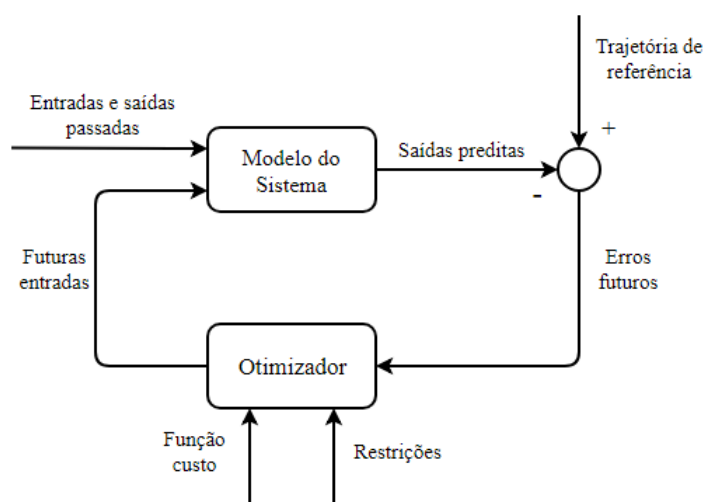


Figura 2.23: Diagrama de blocos da metodologia do MPC, adaptado de (CAMACHO; BORDONS, 2012)

O otimizador (conhecido também como *solver* em inglês) é o responsável pela obtenção da ação de controle ótima a partir das restrições e do valor da função custo do problema. Esse é o bloco mais importante para este trabalho, pois o MPC permite que esse otimizador seja de livre escolha do projetista. Os casos de estudo mostrarão o desempenho do controle quando otimizados pelos algoritmos bioinspirados em estudo. Para entender como é a predição e como funcionam as restrições de um MPC a fundamentação matemática geral será abordada na subseção a seguir.

2.3.2 Fundamentos matemáticos do MPC

Esta subseção apresentará os conceitos e a formalização do MPC mostrados por Alami (2013). Para realizar a predição, o MPC deve saber a dinâmica do sistema e como ele se comporta quando aplicada uma sequência de ações. Esse comportamento pode ser representado pelo modelo em espaço de estados discreto:

$$x(k+1) = f(x(k), \mathbf{u}(k)) = \mathbf{A}x(k) + \mathbf{B}u(k) \quad (2.83)$$

onde, $x \in \mathbb{R}^n$ é o vetor de estados do sistema, $\mathbf{u} \in \mathbb{R}^{n_u}$ é o vetor das ações de controle, $k \in \mathbb{N}$ é o instante de amostragem e $\mathbf{A} \in \mathbb{R}^{n \times n}$ e $\mathbf{B} \in \mathbb{R}^{n \times n_u}$ são matrizes que descrevem o sistema dinâmico do problema. Quando \mathbf{A} e \mathbf{B} são constantes, o sistema é dito linear, já no caso de pelo menos uma das duas matrizes não ser constante o sistema é não-linear e o controle é denominado NMPC (MPC não linear).

A partir do mapeamento de estados na equação (2.83) é possível calcular uma sequência ótima para N (horizonte de predição) ações de controle, formulada como:

$$\tilde{\mathbf{u}}(k) := \begin{pmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{pmatrix} \in \mathbb{R}^{N \cdot n_u} \quad (2.84)$$

Os estados futuros quando $\tilde{\mathbf{u}}(k)$ é aplicado no sistema podem ser representados pela mesma notação:

$$\tilde{\mathbf{x}}(k) := \begin{pmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+N) \end{pmatrix} \in \mathbb{R}^{N \cdot n} \quad (2.85)$$

Baseado na equação (2.83), o cálculo da sequência dos estados futuros dada as ações de controle futuras $\tilde{\mathbf{u}}(k)$ pode ser realizado pela seguinte expressão:

$$\tilde{\mathbf{x}}(k|\tilde{\mathbf{u}}) = X(x(k), \tilde{\mathbf{u}}, \tau, N) \quad (2.86)$$

onde τ é o período de amostragem e X é a função mapeamento dos estados futuros. A Fig. 2.24 ajuda

a verificar como é o funcionamento desse mapeamento.

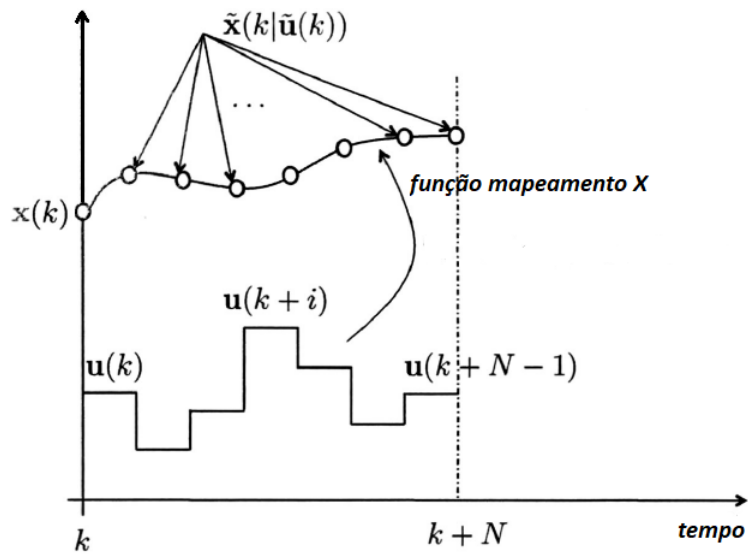


Figura 2.24: Mapeamento dos estados futuros dada uma sequência de ações de controle futura \tilde{u} , adaptado de (ALAMIR, 2013)

Para representar a ação de controle em um determinado instante de tempo específico i , Alamir (2013) apresenta a seguinte notação:

$$u(k+i-1) = \Pi_i^{(n_u, N)} \tilde{u}(k), \quad i \in 1, \dots, N \quad (2.87)$$

onde a matriz de seleção $\Pi_i^{(n_u, N)}$ seleciona o i -ésimo elemento de dimensão n_u de um conjunto de N vetores e pode ser vista na Fig. 2.25.

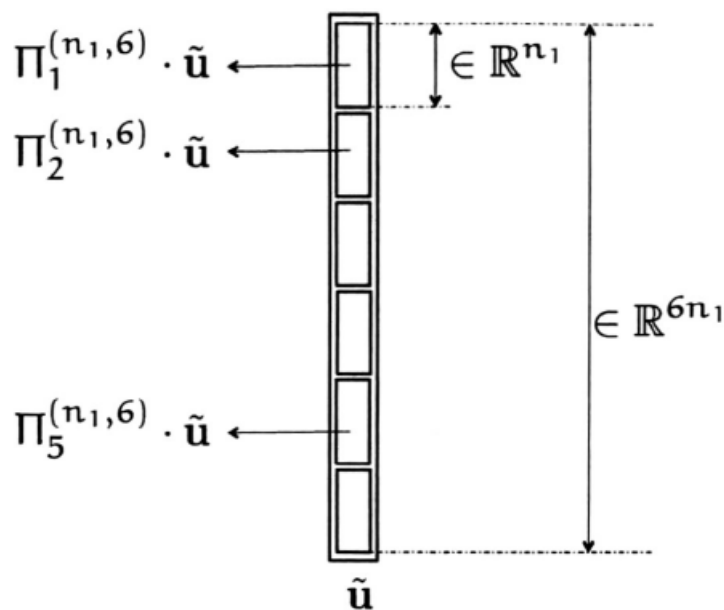


Figura 2.25: Matriz de seleção em um caso especial de $N = 6$ (ALAMIR, 2013)

Da mesma forma, pode-se representar o estado do sistema no instante i como:

$$x(k+i-1) = \Pi_i^{(n,N)} \tilde{x}(k), \quad i \in 1, \dots, N \quad (2.88)$$

O mapeamento dos estados permite definir uma função custo expressa por um escalar e busca avaliar a aptidão do sistema frente à trajetória de referência imposta. Essa função custo depende basicamente das ações de controle $\tilde{u}(k)$ e estados futuros $\tilde{x}(k|\tilde{u})$. O projetista pode definir como é feito o cálculo da função custo e ela estará de acordo com o sistema trabalhado (ALAMIR, 2013). A expressão genérica para a função custo J é:

$$J(\tilde{u}(k), \tilde{x}(k|\tilde{u})) \rightarrow \text{escalar} \quad (2.89)$$

A estratégia do MPC é fornecer a sequência de ações de controle ótima que satisfaçam o melhor valor da função custo e que respeitem as restrições. Tais restrições dependem do problema tratado, mas basicamente podem ser atribuídas aos estados, a ação de controle e à variação da ação de controle entre instantes de amostragem consecutivos. Pode-se generalizar as restrições como uma função g dependente dos estados e da ação de controle.

O melhor valor da função custo normalmente é o menor possível, tornando como tarefa do otimizador resolver um problema de minimização. Dessa forma, pode-se resumir que a metodologia de controle do MPC é da forma:

$$P(x(k)) = \min(J(\tilde{u}|x(k))), \quad \text{dada } g(\tilde{u}|x(k)) \leq 0 \quad (2.90)$$

Assim, a solução do controle é $\tilde{u}^{opt}(x(k))$ que satisfaça a equação (2.90).

2.4 Conclusões do capítulo

As discussões teóricas, apresentadas nas seções anteriores, são relevantes por colocar os principais conceitos utilizados neste trabalho. Com base nesses conceitos, neste trabalho busca-se avaliar os algoritmos bioinspirados como otimizadores para problemas de controle preditivo não-linear (NMPC) do pêndulo invertido (seção 3.1.1) e atitude de satélites (seção 3.1.2). Os algoritmos serão implementados também para resolver as funções da competição CEC2017 (seção 3.1.3).

Os testes estatísticos apresentados neste capítulo são a base para as comparações que serão apresentadas no próximo capítulo. Os testes de normalidade e homogeneidade da variância são úteis na verificação das condições de uso dos testes não-paramétricos. O teste não paramétrico de Friedman (seção 2.2.6) é usado para gerar o *ranking* das meta-heurísticas e juntamente com os procedimentos *post-hoc* serão realizadas comparações 1x1 de cada um dos algoritmos bioinspirados.

Capítulo 3

Metodologia

Este capítulo aborda a metodologia utilizada no trabalho. Serão apresentados os estudos de caso na seção 3.1, a partir desses problemas na seção 3.2 será detalhado como foi feita a aquisição dos dados e quais os parâmetros utilizados nos algoritmos e nos problemas. Por fim, na seção 3.3 pode ser visto a forma com que os desempenhos dos algoritmos foram comparados e quantificados.

3.1 Estudos de caso

Neste trabalho os algoritmos apresentados na seção 2.1 do capítulo anterior são confrontados entre si em relação ao desempenho em três estudos de caso. Dois desses estudos, os algoritmos fazem o papel do otimizador do NMPC e no terceiro eles resolvem 29 problemas matemáticos com restrição de fronteira.

3.1.1 Pêndulo Invertido em um carrinho

O primeiro problema de controle é de inverter e controlar um pêndulo simples na posição invertida ($\theta = 0^\circ$) e o carro na referência ($r = 0$), como ilustrado na Fig. 3.1.

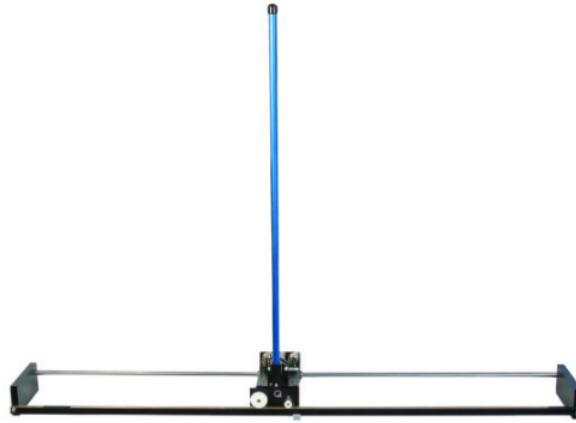


Figura 3.1: Pêndulo invertido da Quanser¹

O modelo dinâmico do pêndulo invertido em um carrinho é não-linear e é obtido pela aplicação das leis de movimento de Newton no carro e no pêndulo, separadamente. O diagrama de blocos da Fig. 3.2 mostra como as forças são aplicadas.

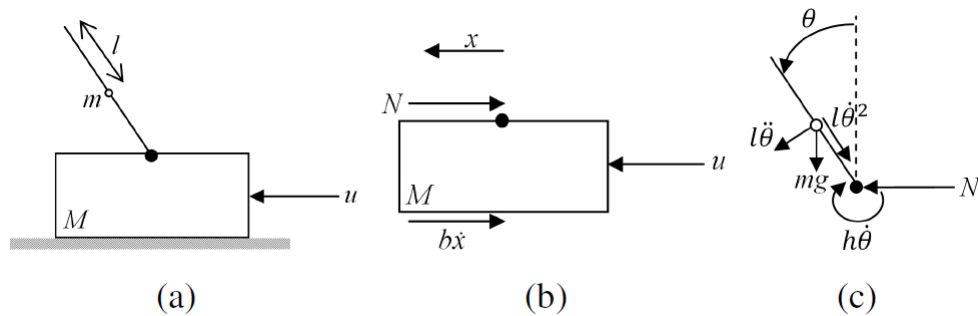


Figura 3.2: a) pêndulo conectado ao carrinho, b) diagrama de corpo livre do carrinho, c) diagrama de corpo livre do pêndulo, (MERCIECA; FABRI, 2011)

O modelamento matemático completo pode ser consultado no Apêndice A do trabalho de Alaniz (2004), porém, em síntese a aceleração do carro e aceleração angular do pêndulo podem ser obtidas pelas equações (3.1) e (3.2).

$$\ddot{x} = \frac{1}{M + m} [u - b\dot{x} - ml\ddot{\theta}\cos(\theta) + ml\dot{\theta}^2\sin(\theta)] \quad (3.1)$$

$$\ddot{\theta} = \frac{3}{4ml^2} [mgl \cdot \sin(\theta) - ml\ddot{x}\cos(\theta) - h\dot{\theta}] \quad (3.2)$$

onde M é a massa do bloco composto pelo carrinho, m é a massa distribuída uniformemente em um pêndulo ideal e $2l$ é o seu comprimento, b é a coeficiente de atrito do carrinho com a superfície, h é o coeficiente de atrito de rotação, u é a força aplicada no bloco, θ é o ângulo entre a normal e o pêndulo e x é

¹<https://www.quanser.com/products/linear-servo-base-unit-inverted-pendulum>

o deslocamento horizontal do bloco. Para melhor representação dos estados do sistema, as equações (3.1) e (3.2) são expressas em termos de x , \dot{x} , θ e $\dot{\theta}$. A equação diferencial de segunda ordem é da forma da equação (3.3) (MERCIECA; FABRI, 2012), o qual χ é um vetor e g é um vetor de campo para combinar os estados em um único vetor.

$$\ddot{\chi} = f(\dot{\chi}, \chi, u), \quad \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} = g(\dot{\chi}, \chi), \quad \chi = \begin{bmatrix} x \\ \theta \end{bmatrix} \quad (3.3)$$

Por fim, a equação é discretizada usando o método de Runge-Kutta de 4ª ordem (ALANIZ, 2004), os vetores K_1 a K_4 são calculados pela equação (3.4).

$$\begin{aligned} K_1 &= f(\dot{\chi}(k), \chi(k), u(k)) \\ K_2 &= f(\dot{\chi}(k) + \frac{T_s}{2} K_1, \chi(k) + \frac{T_s}{2} \cdot \dot{\chi}(k), u(k)) \\ K_3 &= f(\dot{\chi}(k) + \frac{T_s}{2} K_2, \chi(k) + \frac{T_s}{2} \dot{\chi}(k) + \frac{T_s^2}{8} K_1, u(k)) \\ K_4 &= f(\dot{\chi}(k) + T_s K_3, \chi(k) + T_s \dot{\chi}(k) + \frac{T_s^2}{2} K_3, u(k)) \end{aligned} \quad (3.4)$$

Assim, $\dot{\chi}$ e χ para um passo de amostragem é:

$$\begin{aligned} \chi(k+1) &= \chi(k) + T_s [\dot{\chi}(k) + \frac{T_s}{6} (K_1 + K_2 + K_3)] \\ \dot{\chi}(k+1) &= \dot{\chi}(k) + \frac{T_s}{6} (K_1 + 2K_2 + 2K_3 + K_4) \end{aligned} \quad (3.5)$$

Tratando os estados como um vetor $X = [x \ \dot{x} \ \theta \ \dot{\theta}]$, a obtenção da ação de controle é realizada pela otimização de uma função custo dependente dos valores de X (SAMPAIO et al., 2018):

$$\begin{aligned} u_{opt} = \operatorname{argmin} \left[\sum_{i=1}^{N-1} \left(\sum_{j=1}^4 Q_{erro}(X(j, i)) \cdot (X(j, i) - X_{ref}(j, i))^2 Q(j) + R \cdot (u(i) - u_{ref}(i))^2 \right) + \right. \\ \left. + \sum_{j=1}^4 Q_{erro}(X(j, N)) \cdot (X(j, N) - X_{ss}(j))^2 Q_f(j) \right] \end{aligned} \quad (3.6)$$

onde, $X_{ss} = (x = 0, \dot{x} = 0, \theta = 0^\circ, \dot{\theta} = 0^\circ)$ e Q_{erro} da equação (3.6) é uma penalização no valor do estado x do pêndulo caso ele alcance valor maior que $x_{max} = 0,5$ ou menor que $x_{min} = -0,5$ (SAMPAIO et al., 2018). $Q(j)$ e $Q_f(j)$ são as penalizações do erro da trajetória do estado j nos horizontes 1 ao $N-1$ e final N , respectivamente. Apenas o primeiro sinal de controle $u_{opt}(1)$ é aplicado ao sistema no período $[t, t + \tau_s]$ e no instantes seguintes toda a metodologia é aplicada novamente.

3.1.2 Satélite

O segundo problema de controle tem o objetivo controlar o ângulo de atitude de um satélite nos três eixos de rotação considerando o modelo apresentado por Gonzales (2009) de uma plataforma de testes (Fig. 3.3). No modelo utilizado, o elemento responsável pela orientação de cada eixo é uma roda de reação. Essa roda é ligada a um motor que, quando ligado, induz uma velocidade e, pelo princípio da conservação de momento angular, surge um torque de mesma intensidade e direção oposta aplicado no corpo (RODRIGUES; MURILO; SOUZA, 2017).



Figura 3.3: Plataforma de testes: mesa giratória de três eixos, (RODRIGUES; MURILO; SOUZA, 2017)

Denomina-se atitude do simulador a orientação relativa entre o sistema de referência inercial F_i (i_1, i_2, i_3) e o sistema de referência do corpo F_b , referente à parte móvel da plataforma, mas que possui o mesmo centro que F_i .

Para descrever a orientação de F_b com relação a F_i , utiliza-se os ângulos de Euler na sequência de rotações 3-2-1, ou seja, partindo-se de F_i para se chegar a F_b , o eixo 3 deverá ser rotacionado de um ângulo θ_1 , o eixo 2 deverá ser rotacionado de um ângulo θ_2 , e o eixo 1 deverá ser rotacionado de um ângulo θ_3 (RODRIGUES; MURILO; SOUZA, 2017).

O problema trata-se de levar a atitude do corpo, inicialmente de $(0^\circ, 0^\circ, 0^\circ)$, para uma referência imposta $(\theta_1, \theta_2, \theta_3)$. A ação de controle é a velocidade do motor que age em cada roda de reação (u_1, u_2, u_3) .

1. Modelo do sistema:

O modelo matemático da plataforma é não linear e descrito pela equação matricial a seguir (GONZALES, 2009):

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 0 & \sin(\theta_3)/\cos(\theta_2) & \cos(\theta_3)/\cos(\theta_2) \\ 0 & \cos(\theta_3) & -\sin(\theta_3) \\ 1 & \sin(\theta_3)\sin(\theta_2)/\cos(\theta_2) & \cos(\theta_3)\sin(\theta_2)/\cos(\theta_2) \end{bmatrix} \cdot \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (3.7)$$

Os valores de θ_1 , θ_2 e θ_3 são os ângulos de Euler que descrevem a atitude do simulador como a orientação relativa entre o referencial inercial F_i e a referência fixada ao corpo F_b . Já ω_1 , ω_2 e ω_3

são as velocidades angulares da plataforma do eixo de referência F_b em relação a F_i e $\dot{\Omega}_1$, $\dot{\Omega}_2$ e $\dot{\Omega}_3$ são as acelerações das rodas de reação e as variáveis de comando do sistema, por conseguinte, Ω_1 , Ω_2 e Ω_3 são as velocidades das rodas de reação.

I_{11} , I_{22} e I_{33} são os momentos de inércia em torno dos eixos i_1 , i_2 e i_3 . Como dito na seção 2.3 do capítulo anterior, busca-se representar o sistema no modelo de espaço de estados,

$$\dot{X} = f(X, u, A, B) \quad (3.8)$$

onde o vetor de estados é $X = (\theta_1 \ \theta_2 \ \theta_3 \ \omega_1 \ \omega_2 \ \omega_3)^T$ e a ação de controle é $u = (\dot{\Omega}_1 \ \dot{\Omega}_2 \ \dot{\Omega}_3)^T$. A partir das equações e (3.7), as matrizes A e B são definidas,

$$A = \begin{bmatrix} 0 & 0 & 0 & \frac{\sin(\theta_3)}{\cos(\theta_2)} & \frac{\cos(\theta_3)}{\cos(\theta_2)} & \\ 0 & 0 & 0 & 0 & \cos(\theta_3) & -\sin(\theta_3) \\ 0 & 0 & 0 & 1 & \frac{\sin(\theta_3)\sin(\theta_2)}{\cos(\theta_2)} & \frac{\cos(\theta_3)\sin(\theta_2)}{\cos(\theta_2)} \\ 0 & 0 & 0 & 0 & \frac{(I_{22}\omega_3 - I_\omega\Omega_3)}{(I_{11} + I_\omega)} & \frac{(-I_{33}\omega_2 + I_\omega\Omega_2)}{(I_{11} + I_\omega)} \\ 0 & 0 & 0 & \frac{(-I_{11}\omega_3 + I_\omega\Omega_3)}{(I_{22} + I_\omega)} & 0 & \frac{(I_{33}\omega_1 - I_\omega\Omega_1)}{(I_{22} + I_\omega)} \\ 0 & 0 & 0 & \frac{(I_{11}\omega_2 - I_\omega\Omega_2)}{(I_{33} - I_\omega)} & \frac{(I_{22}\omega_1 + I_\omega\Omega_1)}{(I_{33} + I_\omega)} & 0 \end{bmatrix} \quad (3.9)$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{-I_\omega}{I_{11} + I_\omega} & 0 & 0 \\ 0 & \frac{-I_\omega}{I_{22} + I_\omega} & 0 \\ 0 & 0 & \frac{-I_\omega}{I_{33} + I_\omega} \end{bmatrix} \quad (3.10)$$

onde I_ω é a inércia de cada roda de reação. Observa-se que A depende dos valores variáveis de $\Omega_{1,2,3}$ o que torna o sistema não-linear. Para um melhor detalhamento acerca da modelagem do sistema basta consultar (GONZALES, 2009).

2. Entrada degrau filtrada:

Uma estratégia utilizada neste trabalho, apresentada por Rodrigues, Murilo e Souza (2017), foi a aplicação de uma referência filtrada, que é a suavização do sinal :

$$y_i(t) = \theta_i \left(1 - e^{\frac{-3\tau_s}{t_{ref}}}\right) \quad (3.11)$$

onde θ_i é a amplitude do sinal de referência, τ_s é o período de amostragem e t_{ref} o tempo de resposta para o sistema atingir 95% da amplitude θ_i . A Fig. 3.4 mostra a relação entre os dois sinais.

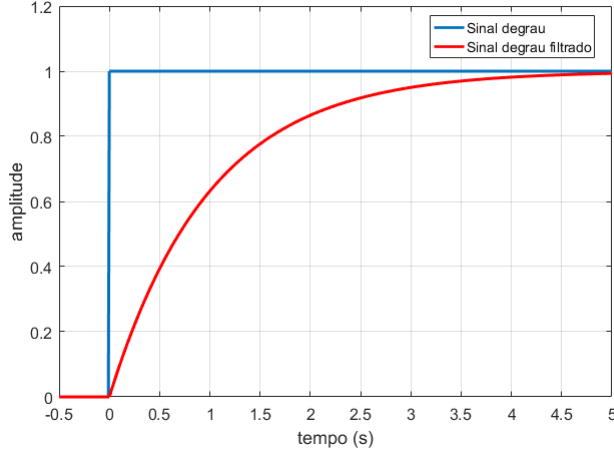


Figura 3.4: Sinais de entrada, degrau e degrau filtrado

A aplicação desse sinal de entrada permite acompanhar a dinâmica do sistema, que não varia rapidamente como em uma entrada degrau e tem o objetivo de reduzir o *overshoot* do sinal de saída.

3. Parametrização:

Outra estratégia utilizada no controle NMPC do Satélite é uma parametrização do sinal de controle criada por Murilo, Alamir e Alberer (2014). No controle preditivo a entrada do sistema (\mathbf{u}) deve 'enxergar' o horizonte de predição (N), ou seja, quanto maior o horizonte maior será a complexidade do problema em definir o \mathbf{u}_{opt} . A fim de diminuir a dimensão da variável de decisão e desacoplá-la da escolha do valor de N , a parametrização de \mathbf{u} é realizada. Ela é representada pela soma de duas exponenciais que modelam o sinal através de dois coeficientes:

$$\mathbf{u}_n(t + i\tau_s) = Sat_{u_{min}}^{u_{max}}(u^* + \alpha_1 \cdot e^{-\lambda i\tau_s} + \alpha_2 \cdot e^{-q\lambda i\tau_s}), \text{ para } t \in [(k-1)\tau_s, k\tau_s[\quad (3.12)$$

onde n é o número de atuadores do sistema, $i \in \{0, \dots, N-1\}$, τ_s é o período de amostragem, $\lambda > 0$ e $q \in \mathbb{N}$ são parâmetros de sintonia, α_1 e $\alpha_2 \in \mathbb{R}^2$ são os coeficientes a serem determinados. Sat é o mapa de saturação $Sat : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ definido como:

$$Sat_{u_{min}}^{u_{max}}(u_i) = \begin{cases} u_{min}^i & \text{se } u_i \leq u_{min}^i \\ u_{max}^i & \text{se } u_i \geq u_{max}^i \\ u_i & \text{caso contrario} \end{cases}, \quad i \in \{1, 2\} \quad (3.13)$$

Considerando três atuadores ($n=3$), o sistema no instante $i = 0$ e omitindo $Sat_{u_{min}}^{u_{max}}$, tem-se:

$$\begin{aligned} u_1(k-1) &= u_1^* + \alpha_1^{u_1} + \alpha_2^{u_1} \\ u_2(k-1) &= u_2^* + \alpha_1^{u_2} + \alpha_2^{u_2} \\ u_3(k-1) &= u_3^* + \alpha_1^{u_3} + \alpha_2^{u_3} \end{aligned} \quad (3.14)$$

onde u_n^* é o valor da variável de controle em estado estacionário. Para $i = 1$ e considerando $K_1 = e^{-\lambda\tau_s}$ e $K_2 = e^{-q\lambda\tau_s}$, tem-se:

$$\begin{aligned}
u_1(k) &= u_1^* + \alpha_1^{u_1} \cdot K_1 + \alpha_2^{u_1} \cdot K_2 \\
u_2(k) &= u_2^* + \alpha_1^{u_2} \cdot K_1 + \alpha_2^{u_2} \cdot K_2 \\
u_3(k) &= u_3^* + \alpha_1^{u_3} \cdot K_1 + \alpha_2^{u_3} \cdot K_2
\end{aligned} \tag{3.15}$$

Devido às restrições da variação da ação de comando i (δ_{max}^i), tem-se:

$$u_i(k) - u_i(k-1) = \alpha_i^{u_i} \cdot (K_1 - 1) + \alpha_2^{u_i} \cdot (K_2 - 1) = \delta_{max}^i \text{ para } i = 1, 2, 3. \tag{3.16}$$

Sabendo-se que o limite inferior é $\delta_{min}^i = -\delta_{max}^i$, a expressão (3.16) pode ser escrita em função de um parâmetro p , o qual $p_j \in [-1, +1]^2$ para $j \in \{1, \dots, n_u\}$.

$$p_j \cdot \delta_{max}^i = \alpha_i^{u_i} \cdot (K_1 - 1) + \alpha_2^{u_i} \cdot (K_2 - 1). \tag{3.17}$$

Definindo a ação de controle em estado estacionário u_n^* como variável a ser encontrada, pode-se escrever que $u_1^* = p_4$, $u_2^* = p_5$ e $u_3^* = p_6$. Organizando as equações (3.15) e (3.16) obtêm-se:

$$\begin{aligned}
u_1(k-1) - p_4 &= \alpha_1^{u_1} + \alpha_2^{u_1} \\
p_1 \cdot \delta_{max}^1 &= \alpha_1^{u_1} \cdot K_1 + \alpha_2^{u_1} \cdot K_2 \\
u_2(k-1) - p_5 &= \alpha_1^{u_2} + \alpha_2^{u_2} \\
p_2 \cdot \delta_{max}^2 &= \alpha_1^{u_2} \cdot K_1 + \alpha_2^{u_2} \cdot K_2 \\
u_3(k-1) - p_6 &= \alpha_1^{u_3} + \alpha_2^{u_3} \\
p_3 \cdot \delta_{max}^3 &= \alpha_1^{u_3} \cdot K_1 + \alpha_2^{u_3} \cdot K_2
\end{aligned} \tag{3.18}$$

O sistema (3.18) é do tipo $M \cdot x = b$, com $x = (\alpha_1^{u_1} \alpha_2^{u_1} \alpha_1^{u_2} \alpha_2^{u_2} \alpha_1^{u_3} \alpha_2^{u_3})$ tem-se:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ K_1 & K_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & K_1 & K_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & K_1 & K_2 \end{pmatrix} \cdot x^T = \begin{pmatrix} u_1(k-1) - p_4 \\ p_1 \cdot \delta_{max}^1 \\ u_2(k-1) - p_5 \\ p_2 \cdot \delta_{max}^2 \\ u_3(k-1) - p_6 \\ p_3 \cdot \delta_{max}^3 \end{pmatrix} \tag{3.19}$$

A matriz b é dependente dos valores de p , logo $M \cdot x = b(p)$, assim a solução do problema é:

$$x(p) = M(K_1, K_2)^{-1} \cdot b(p) \tag{3.20}$$

A expressão (3.20) mostra que os valores de $\alpha_n^{u_n}$, do vetor $x(p)$, depende unicamente dos valores dos parâmetros p da matriz b . Uma vez obtido o valor dos $\alpha_n^{u_n}$, a ação de controle resultado será da forma:

$$\begin{aligned}
\mathbf{u}_1(t + i\tau_s) &= Sat_{u_{min}^{u_{max}}}^{u_{max}}(p_4 + \alpha_1^{u_1}(p) \cdot e^{-\lambda i\tau_s} + \alpha_2^{u_1}(p) \cdot e^{-\mathbf{q} \cdot \lambda i\tau_s}) \\
\mathbf{u}_2(t + i\tau_s) &= Sat_{u_{min}^{u_{max}}}^{u_{max}}(p_5 + \alpha_1^{u_2}(p) \cdot e^{-\lambda i\tau_s} + \alpha_2^{u_2}(p) \cdot e^{-\mathbf{q} \cdot \lambda i\tau_s}) \\
\mathbf{u}_3(t + i\tau_s) &= Sat_{u_{min}^{u_{max}}}^{u_{max}}(p_6 + \alpha_1^{u_3}(p) \cdot e^{-\lambda i\tau_s} + \alpha_2^{u_3}(p) \cdot e^{-\mathbf{q} \cdot \lambda i\tau_s})
\end{aligned} \tag{3.21}$$

Os parâmetros p_j são obtidos por uma função custo a ser otimizada, cuja tarefa de minimização é do algoritmo utilizado no otimizador do NMPC.

$$p_{opt} = \underset{p}{\operatorname{argmin}} \left[\sum_{i=1}^{N-1} \left(\sum_{j=1}^6 Q(j) \cdot (\theta(j, i) - \theta_{ref}(j, i))^2 \right) + \sum_{j=1}^6 Q_f(j) \cdot (\theta(j, N) - \theta_{ss}(j))^2 \right] \quad (3.22)$$

onde $\theta_{ss} = (\theta_1 = 50^\circ, \theta_2 = -30^\circ, \theta_3 = 60^\circ, \omega_1 = 0, \omega_2 = 0, \omega_3 = 0)$ são os valores de referência dos estados, $Q(j)$ e $Q_f(j)$ são as penalizações do erro da trajetória do estado j nos horizontes 1 ao $N - 1$ e final N , respectivamente. Após a solução da equação (3.22), apenas o primeiro sinal de controle $u_i(p_{opt}, 1)$ é aplicado ao sistema no período $[t, t + \tau_s]$ e no instantes seguinte toda a metodologia é aplicada novamente.

3.1.3 Benchmark CEC2017

O último estudo de caso é realizado por um *benchmark* de 29 testes² de otimização numérica de parâmetros reais mono-objetiva da competição CEC2017 com restrição de fronteira apenas (AWAD et al., 2016). Todas as funções são tratadas como problemas de minimização e possuem a mesma fronteira de busca, $[-100, 100]^D$, o qual D é a dimensão tratada. Além disso, o *benchmark* é composto por funções unimodais, multimodais, híbridas e compostas, conforme resumido na Tab. 3.1. As funções básicas que dão origem ao conjunto mostrado são detalhadas no Anexo I. As funções de 1 a 10 são resultado de operações de deslocamento e rotação através do operador $\mathbf{o}_{i1} = [o_{i1}, o_{i2}, \dots, o_{iD}]^T$ e da matriz \mathbf{M}_i . Já as funções híbridas (11 a 20) são formadas pela expressão:

$$F(\mathbf{x}) = g_1(\mathbf{M}_1 \mathbf{z}_1) + g_2(\mathbf{M}_2 \mathbf{z}_2) + \dots + g_N(\mathbf{M}_N \mathbf{z}_N) + F^*(\mathbf{x}) \quad (3.23)$$

onde $g_i(\mathbf{x})$ é a i -ésima função básica usada na construção da função híbrida, N é o número de funções básicas utilizadas e $F^*(\mathbf{x})$ é o valor encontrado na quarta coluna da Tab. 3.1.

²A função 2 foi retirada do *benchmark* pois se mostrava instável, porém a numeração das funções permaneceu inalterada.

Tabela 3.1: *Benchmark* CEC2017

	No	Função	$F_i^* F_i(x^*)$
Funções unimodais	1	Função Bent Cigar deslocada e rotacionada	100
	3	Função Zakharov deslocada e rotacionada	300
Funções multimodais simples	4	Função de Rosenbrock deslocada e rotacionada	400
	5	Função de Rastrigin deslocada e rotacionada	500
	6	Função de Shaffer expandida, deslocada e rotacionada	600
	7	Função Lunacek Bi_Rastrigin deslocada e rotacionada	700
	8	Função não contínua de Rastrigin deslocada e rotacionada	800
	9	Função Levy deslocada e rotacionada	900
	10	Função de Schwefel deslocada e rotacionada	1000
Funções híbridas	11	Função híbrida 1 (N=3)	1100
	12	Função híbrida 2 (N=3)	1200
	13	Função híbrida 3 (N=3)	1300
	14	Função híbrida 4 (N=4)	1400
	15	Função híbrida 5 (N=4)	1500
	16	Função híbrida 6 (N=4)	1600
	17	Função híbrida 7 (N=5)	1700
	18	Função híbrida 8 (N=5)	1800
	19	Função híbrida 9 (N=5)	1900
	20	Função híbrida 10 (N=6)	2000
Funções compostas	21	Função composta 1 (N=3)	2100
	22	Função composta 2 (N=3)	2200
	23	Função composta 3 (N=4)	2300
	24	Função composta 4 (N=4)	2400
	25	Função composta 5 (N=5)	2500
	26	Função composta 6 (N=5)	2600
	27	Função composta 7 (N=6)	2700
	28	Função composta 8 (N=6)	2800
	29	Função composta 9 (N=3)	2900
	30	Função composta 10 (N=3)	3000

Por fim, as funções 21 a 30 são compostas e seguem a forma:

$$F(\mathbf{x}) = \sum_{i=1}^N (\omega_i \cdot [\lambda_i g_i(\mathbf{x}) + bias_i]) + F^* \quad (3.24)$$

onde $bias_i$ define qual será o ótimo global, λ_i é usado para controlar a altura da função g_i e w_i é o seu peso:

$$w_i = \frac{1}{\sqrt{\sum_{j=1}^D (x_j - o_{ij})^2}} \exp\left(-\frac{\sum_{j=1}^D (x_j - o_{ij})^2}{2D\sigma_i^2}\right) \quad (3.25)$$

e então $\omega_i = w_i / \sum_{i=1}^n w_i$.

A composição total das funções pode ser vista no Anexo II. Para informações adicionais o arquivo original da competição está disponível em (AWAD et al., 2016).

3.2 Método de aquisição de dados

A partir dos problemas apresentados na seção anterior os dados de desempenho das meta-heurísticas são coletados e confrontados entre si. Nesta seção, será apresentada a metodologia de aquisição desses dados e os parâmetros pertinentes a cada problema. A partir daqui a sigla utilizada para o controle preditivo é NMPC, devido aos modelos não lineares apresentados como estudos de caso.

3.2.1 Estratégia otimizador NMPC

Os dois problemas de controle NMPC são semelhantes em termos da estratégia empregada. O controle funciona na obtenção da ação de controle ótima que apresente a menor função custo do problema. A função custo é calculada ao longo de um horizonte de predição N , porém, apenas o primeiro elemento ($u_{opt}(1)$) é aplicado ao sistema e no seguinte instante de amostragem todo o controle é realizado novamente.

A estratégia de controle aplicada ao algoritmo do otimizador NMPC faz parte do estudo realizado por Sampaio et al. (2018) e nele foram empregados dois métodos: o primeiro, proposto por (SIVANANAITHA-PERUMAL et al., 2007) refere-se à inicialização dos agentes de busca a cada tempo de amostragem do otimizador. Como o pêndulo e o satélite são sistemas dinâmicos considerados lentos e seus estados não sofrem bruscas variações de um instante de decisão a outro subsequente, o resultado que o otimizador obteve para o instante k ($u_{opt}(k)$) é um dos agentes de busca da população inicial da meta-heurística no instante $k + 1$.

A segunda estratégia é a inicialização de outro indivíduo da população com o valor de \mathbf{u}_{ss} , isso serve para que ao final do controle haja a estabilização da ação de controle, já que os indivíduos tendem a convergir para \mathbf{u}_{ss} ao final da otimização.

O otimizador, através dos algoritmos bioinspirados, realiza o processo mostrado pela Fig. 3.5.

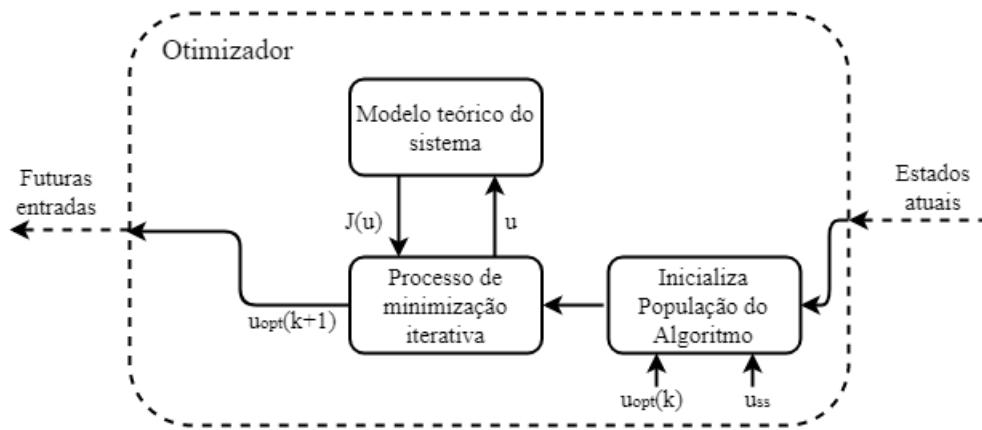


Figura 3.5: Processo interno do otimizador realizado pelo algoritmo bioinspirado

O bloco do processo de minimização iterativa consiste na atualização da população e busca pelo valor u_{opt} que satisfaça o menor custo J . As condições iniciais dos sistemas são mostrados na Tab. 3.2.

Tabela 3.2: Condições iniciais - sistemas NMPC

Pêndulo	
estado	valor inicial
(x, \dot{x})	$(0,0)$
$(\theta, \dot{\theta})$	$(180^\circ, 0^\circ)$
Satélite	
estado	valor inicial
(θ_1, ω_1)	$(0^\circ, 0^\circ)$
(θ_2, ω_2)	$(0^\circ, 0^\circ)$
(θ_3, ω_3)	$(0^\circ, 0^\circ)$

3.2.2 Parâmetros

Em todos os estudos de caso foram definidos parâmetros de simulação específicos. Para o controle do pêndulo invertido foram utilizados os parâmetros definidos no trabalho do Sampaio et al. (2018) e são mostrados na Tab. 3.3. Já no controle do satélite os parâmetros são fundamentados pelo trabalho do Rodrigues (2018), conforme indicados na Tab. 3.4.

Tabela 3.3: Parâmetros do sistema do Pêndulo

	Parâmetro	Valor
Meta-heurísticas	S	10
	Max_{iter}	30
Pêndulo	m	7,3 [Kg]
	M	14,6 [Kg]
	g	9,81 [m/s^2]
	l	1,2 [m]
	b	14,6 [Kg/s]
h	0,0136 [Kg · m^2/s]	
NMPC	N	20
	N_u	20
	τ_s	0,1 [s]
	Q	[10^4 1 10^4 1]
	Q_f	[10^3 10^3 10^3 10^3]
	R	1
	u_{min}	-45 [N]
	u_{max}	45 [N]
δ_u	50 [N]	

Tabela 3.4: Parâmetros do sistema do Satélite

	Parâmetro	Valor
Meta-heurísticas	S	4
	Max_{iter}	15
Satélite	I_ω	$1,8 \cdot 10^{-3}$ [Kg · m^2]
	I_{11}	1,17 [Kg · m^2]
	I_{22}	1,17 [Kg · m^2]
	I_{33}	1,17 [Kg · m^2]
NMPC	N	75
	τ_s	0,1 [s]
	Q	(10^2 10^2 10^2 1 1 1)
	Q_f	1
	λ	0,1
	q	8,0
	u_{min}	(-1,5 -1,5 -1,5) [N]
	u_{max}	(1,5 1,5 1,5) [N]
	δ_u	(1 1 1) [N]
	p_{min}	(-1 -1 -1)
	p_{max}	(1 1 1)
	t_{ref}	30 [s]

Já as funções do *benchmark* CEC2017 foram executadas pelas meta-heurísticas com parâmetros da Tab. 3.5 baseados no artigo Awad et al. (2016). O número máximo de iterações (Max_{iter}) foi restringido a 1000 para verificar o comportamento das meta-heurísticas em poucas iterações.

Tabela 3.5: Parâmetros do *benchmark* CEC2017

	Parâmetro	Valor
Meta-heurísticas	S	20
	Max_{iter}	1000
Funções	D	10/30/100
	l_b	-100
	u_b	100

Algumas meta-heurísticas possuem parâmetros a sintonizar, para o PSO aplicado ao NMPC os valores foram determinados no trabalho Sampaio et al. (2018). As demais meta-heurísticas foram implementadas com os parâmetros indicados no Capítulo 2. O PSO é configurado com os seguintes valores:

Tabela 3.6: Parâmetros PSO

	Parâmetro	Pêndulo	Satélite	CEC2017
PSO	c_1	2,1	2,1	2,1
	c_2	1	0,7	2,1
	w_0	0,9	0,9	0,9
	w_f	0,4	0,1	0,2
	V_{max}	0,1	0,1	10

3.2.3 Geração dos resultados

Para permitir a análise de desempenho das meta-heurísticas, o número de problemas é limitado pela regra de ouro (*rule of thumb* em inglês), a qual diz que em comparações múltiplas de k meta-heurísticas perde-se sua significância quando a quantidade inadequada de problemas n é escolhida.

Assim, defini-se esse valor no intervalo $2 \cdot k \leq n < 8 \cdot k$ (GARCÍA et al., 2009; GARCÍA et al., 2009). Sendo assim, os estudos do pêndulo e do *benchmark* CEC2017, com 10 meta-heurísticas em análise, estão limitados ao número $20 \leq n < 80$ de problemas e o estudo do satélite está entre 22 e 88 problemas já que são 11 meta-heurísticas na comparação.

Problemas, neste trabalho, são definidos como a quantidade de funções do *benchmark* da competição e, no caso do pêndulo, como os instantes de amostragem pelos quais o controle é aplicado ou, para o satélite, a mediana desses instantes. Esses problemas serão detalhados mais a frente.

O número de execuções das meta-heurísticas foi estabelecido como indicado em (AWAD et al., 2016) (51 vezes) e a extração das informações dessas execuções foi definida como a mediana desses valores. Pois segundo Demšar (2006), as médias são susceptíveis aos dados isolados e elas podem permitir um bom desempenho em um conjunto de dados para compensar o mau desempenho geral da meta-heurística, ou o contrário. Essa orientação é seguida por Derrac et al. (2011) o qual recomenda o uso das medianas na comparação de múltiplas meta-heurísticas.

(a) Pêndulo invertido:

A simulação tem um tempo total de 20 segundos e, como dito, um tempo de amostragem $\tau_s = 0,1s$, resultando em 200 problemas de otimização.

Como será verificado no capítulo seguinte, mais da metade do tempo de simulação é usado pelo controlador para erguer o pêndulo e o valor da função custo nestes instantes não reflete o real desempenho do otimizador. Sendo assim, foram definidas 61 problemas dentro do controle, sendo eles cada instante de decisão entre 14° e o 20° segundo de simulação. O controle foi executado 51 vezes para cada meta-heurística como otimizadora do NMPC. Em cada tarefa foi extraída a mediana e elaborada a tabela para análise dos resultados.

(b) Satélite:

O sistema do satélite foi simulado em um tempo de 100 segundos, definidos assim 1000 instantes de decisão para $\tau_s = 0,1s$. Por se tratar de um sistema, cuja trajetória de deslocamento é importante, foram definidas 62 tarefas extraíndo a média de cada 16 tempos de amostragem consecutivos até o tempo de 99,2 segundos. A referência para os ângulos de Euler do sistema é $\theta_1 = +50^\circ$, $\theta_2 = -30^\circ$ e $\theta_3 = +60^\circ$ (RODRIGUES, 2018).

Da mesma forma que o controle do pêndulo, o controle do satélite foi simulado 51 vezes e o resultado das tarefas foi extraído da mediana dessas simulações.

(c) **CEC2017:**

Assim como os problemas NMPC, o *benchmark* CEC2017 foi resolvido pelas meta-heurísticas 51 vezes e para cada função foi extraída a mediana dos resultados e tabulados para a análise de desempenho.

3.3 Método de análise de desempenho

A partir dos resultados tabelados de cada meta-heurística, os primeiros testes são acerca da normalidade e homogeneidade da variância. Conforme apresentados na seção 2.2.4 e 2.2.5, são realizados 3 testes de normalidade e o teste de Levene para homogeneidade da variância.

Verificadas as condições de normalidade e homogeneidade da variância dos dados, partindo que a independência dos mesmos é provada por não serem obtidos por meta-heurísticas distintas em instantes diferentes, parte-se para o uso dos métodos não-paramétricos de comparação. O *software* KEEL é alimentado e retorna o *ranking* de Friedman. Através do *valor-p* é possível descobrir se existe pelo menos duas meta-heurísticas não familiares.

Após o *ranking* de Friedman, são aplicados os testes *post-hoc*, cujo resultado da comparação de todos os pares de meta-heurísticas possíveis verifica a familiaridade entre eles. A meta-heurística no topo do *ranking* e seus familiares são consideradas aquelas com o melhor resultado para o problema proposto.

Para o problema de controle do satélite, a comparação é realizada incluindo o desempenho do otimizador SQP³ (*Sequential Quadratic Programming* em inglês) utilizado por Rodrigues (2018).

Para complementar o estudo, são feitas análises qualitativas, indicando os estados e ações de controle dos problemas NMPC. Além de abordar de uma forma mais visual os resultados obtidos com as funções da competição CEC2017.

3.4 Modificações e adaptações

As meta-heurísticas apresentadas neste trabalho incluem as versões clássicas, PSO e DE por exemplo, e outros oriundos de variações, como o JADE e o LSHADE (baseados no DE). A fim de melhorar o desempenho já obtido, serão apresentadas modificações e adaptações dos algoritmos GWO e MFO no capítulo a seguir na seção 4.5.

³Método de iteração que utiliza funções quadráticas para aproximar a função custo em uma determinada região.

3.5 Método de análise do tempo de execução

O presente trabalho foi desenvolvido em ambiente de simulação utilizando o *software* MatLab 2016a, dessa forma, sabe-se que os códigos não estão estruturados para execução em tempo real (o que demanda uma linguagem mais sofisticada). Mesmo assim, nesta seção será apresentada uma metodologia simples de comparação do tempo de execução das meta-heurísticas utilizadas com o objetivo de verificar grandes divergências entre esses tempos.

Para o desempenho nas funções da competição o método foi baseado no procedimento proposto por Awad et al. (2016):

1. Avalia a função 18 para 1000 iterações em cada dimensão proposta, executar por 5 vezes e extrair a média $\bar{T}_1 = mean(T_1)$.
2. A análise é então realizada através da razão \bar{T}_1/T_0 . O qual T_0 tem o mesmo valor de \bar{T}_1 da meta-heurística com o melhor ranqueamento no desempenho da solução.

Da mesma forma, nos problemas do pêndulo e satélite o valor de T_0 será a média do tempo de execução em cada período de amostragem considerando a simulação da meta-heurística com o melhor *ranking* no método de Friedman e para as demais meta-heurísticas denominado de T_1 . Igualmente, a razão T_1/T_0 será usada para analisar o tempo de execução de cada meta-heurística. Assim, a meta-heurística com melhor *ranking* terá razão igual a 1, sendo o referencial para o restante do grupo.

3.6 Conclusões do capítulo

Neste capítulo foram apresentadas as etapas que discernem a metodologia proposta. Baseados na livre distribuição dos dados resultantes, os testes não-paramétricos mostram-se promissores na computação natural, já que muitas das meta-heurísticas encontradas nela são provenientes de métodos estocásticos. Além disso, as funções da competição CEC2017 fazem parte de um grupo de métodos clássicos de comparação do desempenho de meta-heurísticas e o uso de algoritmos bioinspiradas como otimizadores do NMPC vem crescendo, já que mostram ser tão eficientes quanto os modelos analíticos já aplicados.

No capítulo a seguir serão mostrados os resultados obtidos aplicando a metodologia apresentada.

Capítulo 4

Resultados

Neste capítulo são apresentados os resultados obtidos no trabalho. Iniciando com os testes de normalidade (seção 4.2) e seguindo para os de homogeneidade de variância (seção 4.3). A comparação das meta-heurísticas é mostrada na seção 4.4, no qual encontra-se, também, a análise qualitativa, que busca analisar o comportamento dos estados e ações de comando dos problemas de controle quando aplicados os algoritmos bioinspirados. As meta-heurísticas modificadas e adaptadas são mostradas na seção 4.5. Por fim, a análise comparativa do tempo de execução das meta-heurísticas é mostrada na seção 4.7.

4.1 Resultados dos estudos de caso

Os dados resultantes dos estudos de caso podem ser vistos nas tabelas apresentadas no Apêndice A. As mesmas mostram a mediana para cada amostra, sendo cada amostra composta de 51 resultados experimentais.

4.2 Testes de normalidade

Os testes de normalidade associam aos dados um *valor-p*, o qual indica quanto a população (representada pela amostra) se assemelha a uma distribuição normal, como já mencionado na seção 2.2.4. Vale lembrar que, para $\text{valor-p} > \alpha$, H_0 é aceita como verdadeira e indica normalidade da população, por conseguinte para $\text{valor-p} \leq \alpha$, H_0 é rejeitada e indica não-normalidade da população. Para $\alpha = 0,05$ os resultados são mostrados nas tabelas a seguir. Os valores em negrito indicam a normalidade dos dados.

(a) **Pêndulo**

Tabela 4.1: Resultado: testes de normalidade - Pêndulo

	KS-L		SW		DAP	
	D_n	valor-p	Z	valor-p	DAP	valor-p
PSO	0,283	0,000	0,685	0,000	33,358	0,000
DE	0,287	0,000	0,626	0,000	52,299	0,000
DA	0,351	0,000	0,440	0,000	79,502	0,000
SSA	0,070	0,200	0,976	0,276	1,192	0,551
WOA	0,248	0,000	0,768	0,000	416,173	0,000
MFO	0,354	0,000	0,428	0,000	77,988	0,000
GWO	0,367	0,000	0,384	0,000	89,132	0,000
SCA	0,418	0,000	0,433	0,000	54,726	0,000
JADE	0,138	0,006	0,901	0,000	5,672	0,059
LSHADE	0,255	0,000	0,792	0,000	11,042	0,004

De acordo com o resultado mostrado na Tab. 4.1 a meta-heurística SSA foi considerada normal para os 3 testes realizados e o JADE pelo teste DAP.

(b) **Satélite**

Tabela 4.2: Resultado: testes de normalidade - Satélite

	KS-L		SW		DAP	
	D_n	valor-p	Z	valor-p	DAP	valor-p
PSO	0,416	0,000	0,494	0,000	48,425	0,000
DE	0,414	0,000	0,495	0,000	48,390	0,000
DA	0,417	0,000	0,492	0,000	48,599	0,000
SSA	0,412	0,000	0,496	0,000	48,477	0,000
WOA	0,415	0,000	0,495	0,000	48,686	0,000
MFO	0,415	0,000	0,494	0,000	48,491	0,000
GWO	0,416	0,000	0,493	0,000	48,529	0,000
SCA	0,416	0,000	0,493	0,000	48,570	0,000
JADE	0,417	0,000	0,492	0,000	48,586	0,000
LSHADE	0,417	0,000	0,491	0,000	48,595	0,000
SQP	0,363	0,000	0,613	0,000	33,917	0,000

Na Tab. 4.2 pode-se verificar que os dados de todas as meta-heurísticas foram considerados não-normais.

(c) **CEC2017**

Tabela 4.3: Resultado: testes de normalidade - CEC2017 com dimensão 10

	KS-L		SW		DAP	
	D_n	valor-p	Z	valor-p	DAP	valor-p
PSO	0,325	0,000	0,679	0,000	28,536	0,000
DE	0,473	0,000	0,214	0,000	69,729	0,000
DA	0,494	0,000	0,192	0,000	70,144	0,000
SSA	0,501	0,000	0,252	0,000	65,493	0,000
WOA	0,482	0,000	0,240	0,000	68,018	0,000
MFO	0,458	0,000	0,247	0,000	68,050	0,000
GWO	0,458	0,000	0,361	0,000	44,799	0,000
SCA	0,504	0,000	0,189	0,000	70,174	0,000
JADE	0,083	0,200	0,961	0,353	3,413	0,182
LSHADE	0,080	0,200	0,964	0,410	3,076	0,215

Tabela 4.4: *Valor-p* dos testes de normalidade - CEC2017 com dimensão 30

	KS-L		SW		DAP	
	D_n	valor-p	Z	valor-p	DAP	valor-p
PSO	0,490	0,000	0,310	0,000	49,016	0,000
DE	0,525	0,000	0,186	0,000	70,195	0,000
DA	0,484	0,000	0,206	0,000	69,865	0,000
SSA	0,473	0,000	0,359	0,000	47,814	0,000
WOA	0,485	0,000	0,218	0,000	69,228	0,000
MFO	0,526	0,000	0,186	0,000	70,196	0,000
GWO	0,500	0,000	0,192	0,000	70,143	0,000
SCA	0,466	0,000	0,230	0,000	68,909	0,000
JADE	0,340	0,000	0,512	0,000	42,903	0,000
LSHADE	0,383	0,000	0,423	0,000	55,169	0,000

Tabela 4.5: *Valor-p* dos testes de normalidade - CEC2017 com dimensão 100

	KS-L		SW		DAP	
	D_n	valor-p	Z	valor-p	DAP	valor-p
PSO	0,466	0,000	0,307	0,000	48,334	0,000
DE	0,492	0,000	0,234	0,000	67,887	0,000
DA	0,444	0,000	0,281	0,000	63,374	0,000
SSA	0,475	0,000	0,352	0,000	46,886	0,000
WOA	0,471	0,000	0,253	0,000	66,096	0,000
MFO	0,453	0,000	0,277	0,000	63,137	0,000
GWO	0,473	0,000	0,248	0,000	66,852	0,000
SCA	0,423	0,000	0,323	0,000	56,353	0,000
JADE	0,444	0,000	0,251	0,000	67,578	0,000
LSHADE	0,507	0,000	0,193	0,000	70,107	0,000

De acordo com os *valores-p* mostrados nas Tabelas 4.3 a 4.5 apenas os dados de JADE e LSHADE foram considerados normais para os três testes.

4.3 Testes de homogeneidade da variância

No teste de Levene (vide seção 2.2.5), cada meta-heurística é tratada como um grupo (amostra), sendo verificada a homogeneidade da variância entre eles (k grupos). Dessa forma, os dados de entrada são os mesmos do Apêndice A. A estatística de teste é \mathbf{W} e seu valor é utilizado para a obtenção do *valor-p* pela função $f_{cdf}(\mathbf{W}, k-1, N-k)$ do MatLab. Aceita-se H_0 caso o *valor-p* $> \alpha$; ou seja, para um nível de significância α tem-se que as meta-heurísticas possuem mesma variância.

O teste de Levene para cada estudo de caso teve os seguintes resultados:

(a) **Pêndulo**

Tabela 4.6: Variância das meta-heurísticas - Pêndulo

Meta-heurística	Variância ($\cdot 10^3$)
PSO	0,0902
DE	0,0900
DA	11665,9742
SSA	0,2615
WOA	0,0056
MFO	56260,5797
GWO	4473,7441
SCA	0,3847
JADE	0,0114
LSHADE	0,0017
Tamanho amostra	61

O cálculo resultou em $W = 247,5575$ para uma distribuição F com 9 e 600 graus de liberdade e valor- $p = 0,00$. Neste caso, as meta-heurísticas possuem variâncias diferentes. Já que valor- $p \leq 0,05$, H_0 é rejeitado ao nível de significância de 0,05.

(b) **Satélite**

Tabela 4.7: Variância das meta-heurísticas - Satélite

Meta-heurística	Variância ($\cdot 10^3$)
PSO	0,0287
DE	0,1119
DA	0,1120
SSA	0,1130
WOA	0,1134
MFO	0,1125
GWO	0,1140
SCA	0,1174
JADE	0,1121
LSHADE	0,1151
SQP	0,1158
Tamanho amostra	62

O valor encontrado para W foi de 0,9458. O valor- p resultante é 0,48. Isso indica que as meta-heurísticas possuem resultados com variâncias iguais para um $\alpha = 0,05$, pois aceita-se como verdadeira a hipótese nula (valor- $p > \alpha$).

(c) **CEC2017**

Tabela 4.8: Variância das meta-heurísticas - CEC2017

D = 10		D = 30		D = 100	
Meta-heurística	Variância ($\cdot 10^{10}$)	Meta-heurística	Variância ($\cdot 10^{10}$)	Meta-heurística	Variância ($\cdot 10^{15}$)
PSO	0,0001	PSO	0,0153	PSO	461,7405
DE	0,0001	DE	0,0056	DE	0,0053
DA	3347640,6600	DA	1376576727,6797	DA	1747633,6281
SSA	1,7634	SSA	34207522,2159	SSA	15276,1143
WOA	1,5923	WOA	406417968,6730	WOA	739931,9438
MFO	1142,6544	MFO	33129405,7862	MFO	295346,4104
GWO	10,1052	GWO	760,8602	GWO	46,7281
SCA	431812,0795	SCA	482090152,9561	SCA	1021916,2651
JADE	2,1705	JADE	3811136,6827	JADE	99229,1278
LSHADE	0,0013	LSHADE	2,4200	LSHADE	0,5565
Tamanho amostra	29	Tamanho amostra	29	Tamanho amostra	29

A estatística W para cada análise foi de: 3,9673 ($D = 10$), 4,34 ($D = 30$) e 4,3451 ($D = 100$), assim, os *valores-p* encontrados pelo teste de Levene para 9 e 280 graus de liberdade tiveram aproximadamente o mesmo valor de 0,001 nas três dimensões. Portanto, para os problemas do *benchmark* CEC2017 nas dimensões 10, 30 e 100, as meta-heurísticas não possuem mesma variância.

4.4 Análise de múltiplas meta-heurísticas

Tendo visto que não foram comprovados os requisitos de normalidade e homogeneidade dos dados para o mesmo problema é **recomendável o uso da comparação por métodos não paramétricos** (vide seção 2.2.6), começando pela análise das meta-heurísticas por meio do ranqueamento proposto por Friedman. Com o uso da ferramenta de software KEEL, os *rankings* das meta-heurísticas para os dois problemas de controle podem ser vistos na Tab. 4.9 e para o benchmark CEC2017 na Tab. 4.10.

Tabela 4.9: *Ranking* das meta-heurísticas - NMPC

Pêndulo		Satélite	
Meta-heurística	Ranking	Meta-heurística	Ranking
MFO	1,26	JADE	3,32
PSO	2,54	MFO	3,62
GWO	2,63	LSHADE	3,66
DE	4,34	GWO	4,69
DA	4,54	DA	5,00
LSHADE	5,80	PSO	6,09
JADE	7,19	DE	6,19
SCA	7,67	SCA	7,48
WOA	9	SSA	8,03
SSA	10	WOA	8,50
-	-	SQP	9,38
χ_f	514,67	χ_f	254,01
<i>valor-p</i> (Friedman)	$1,87 \cdot 10^{-10}$	<i>valor-p</i> (Friedman)	$1,26 \cdot 10^{-10}$
F_f	899,62	F_f	42,33
<i>valor-p</i> (IeD)	$4,44 \cdot 10^{-16}$	<i>valor-p</i> (IeD)	$1,60 \cdot 10^{-63}$

De acordo com os *valores-p* mostrados na Tab. 4.9, para cada problema NMPC, ao menos um das meta-heurísticas não pertence a mesma família que os outros, pois rejeita-se H_0 , lembrando que *valor-p* $\leq 0,05$ indica rejeição da hipótese nula H_0 (vide subseção 2.2.2).

Tabela 4.10: *Ranking* das meta-heurísticas - CEC2017

D=10		D=30		D=100	
Meta-heurística	Ranking	Meta-heurística	Ranking	Meta-heurística	Ranking
JADE	1,43	JADE	1,41	JADE	1,96
LSHADE	2,27	LSHADE	2,13	LSHADE	2,62
DE	3,68	DE	3,82	PSO	2,93
PSO	4,62	PSO	3,86	SSA	4,24
SSA	4,98	SSA	5,13	DE	4,89
GWO	5,68	GWO	5,24	GWO	4,96
MFO	6,65	MFO	6,86	MFO	7,24
SCA	7,75	WOA	8,51	WOA	7,68
WOA	8,51	SCA	8,75	DA	9,20
DA	9,37	DA	9,24	SCA	9,24
χ_f	195,81	χ_f	219,09	χ_f	206,21
<i>valor-p</i> (Friedman)	$1,05 \cdot 10^{-10}$	<i>valor-p</i> (Friedman)	$1,22 \cdot 10^{-10}$	<i>valor-p</i> (Friedman)	$9,78 \cdot 10^{-11}$
F_f	84,10	F_f	146,39	F_f	105,39
<i>valor-p</i> (IeD)	$4,44 \cdot 10^{-16}$	<i>valor-p</i> (IeD)	$2,22 \cdot 10^{-16}$	<i>valor-p</i> (IeD)	0,0

Os valores- p obtidos pelo teste de Friedman e de Iman e Davenport são menores que 0,05 e dessa forma pode-se concluir que, equitativamente com os problemas de controle, H_0 é rejeitada; então para o *benchmark* CEC2017, nas três dimensões testadas, há pelo menos uma meta-heurística que não pertence à mesma família dos demais.

A comparação entre pares das meta-heurísticas, utilizando o *ranking* de Friedman resulta em um *valor-p*, o qual a um nível de significância α indica se ambos são da mesma família. Esse valor é denominado de *valor-p* não ajustado (DERRAC et al., 2011), pois é resultado da análise de duas meta-heurísticas, sem levar em consideração o restante do grupo. Para realizar uma comparação levando em conta todos as meta-heurísticas, os *valores-p* ajustados (APVs) são calculados pelos procedimentos *post-hoc*. Esses APVs são denominados como: (a) p_{Neme} (Nemenyi) e (b) p_{Shaf} (Shaffer). Os resultados podem ser vistos nas tabelas do Apêndice B.

Os resultados da comparação múltipla, para um nível de significância de $\alpha = 0,05$, são:

(a) **Pêndulo:**

Considerando o resultado encontrado na Tab. B.1 tem-se que o *valor-p* não ajustado aceita as hipóteses das linhas 43 a 45 como verdadeiras, já que $\text{valor-p} > \alpha$, ou seja, as duas meta-heurísticas de cada hipótese são considerados provenientes da mesma família, portanto com desempenhos semelhantes. Assim, a meta-heurística MFO é o que possui o melhor desempenho já que tem o melhor *ranking* e não pertence a família de outra meta-heurística.

Com a aplicação do procedimento Nemenyi (vide seção 2.2.7) tem-se que as hipóteses das linhas 35 a 45 são aceitas, por conseguinte, pelas hipóteses das linhas 38 e 40 as meta-heurísticas PSO e GWO também possuem o melhor desempenho já que pertencem à mesma família do MFO. Pelo procedimento Shaffer (vide seção 2.2.7), aceitam-se as hipóteses das linhas 36 a 45. Portanto, mantém o resultado do procedimento Nemenyi como MFO, PSO e GWO como as meta-heurísticas com os melhores desempenhos para o problema de controle do pêndulo invertido.

(b) **Satélite:**

Aplicando a mesma análise feita no problema do pêndulo e verificando o resultado mostrado na Tab. B.2 tem-se que o *valor-p* não ajustado indica que as hipóteses das linhas 44 a 55 são aceitas, e pelos resultados dos testes das linhas 51 e 52 as meta-heurísticas LSHADE e MFO são da mesma família que o JADE e, portanto, são as três meta-heurísticas com os melhores desempenhos.

Pelos procedimentos *post-hoc* têm-se: p_{Neme} é maior que α para as hipóteses das linhas 31 a 55, julgando-as como verdadeiras; já, Shaffer julgou como verdadeiras as hipóteses das linhas 34 a 55, pois $p_{Shaf} > \alpha$. Assim, além de MFO e LSHADE, as meta-heurísticas GWO e DA, pelas hipóteses das linhas 38 e 34, também possuem o mesmo desempenho (família) que o JADE, incluindo-os no grupo de meta-heurísticas com os melhores desempenhos.

Pode-se analisar que o uso dos algoritmos bioinspirados como otimizador do método NMPC superou o desempenho do SQP (RODRIGUES, 2018) pelo *ranking* de Friedman. Por outro lado, pelos valores- p não ajustados WOA teve o mesmo desempenho do SQP (por p_{Shaf} inclui-se SSA nessa lista e ainda por p_{Neme} outra meta-heurística que teve o mesmo desempenho do SQP foi o SCA).

(c) **CEC2017:**

O resultado mostrado na Tab. B.3 indica que os valores- p não ajustados aceitam as hipóteses para os testes das linhas 35 a 45, Nemenyi e Shaffer julgam como verdadeiras as hipóteses das linhas 27 a 45. Assim, o não ajuste dos valores- p indica que JADE e LSHADE possuem o melhor desempenho (hipótese da linha 42) e pelos procedimentos *post-hoc* a meta-heurística DE também está dentro dessa família.

Para as funções com dimensão 30 (Tab. B.4) as hipóteses verdadeiras são das linhas 36 a 45, para o *valor-p* não ajustado); das linhas de 27 a 45, para o procedimento Nemenyi e linhas de 30 a 45, para o procedimento Shaffer. Portanto, para todos os casos, JADE e LSHADE são da mesma família e possuem o melhor desempenho (vide teste da linha 41).

Por fim, na Tab. B.5 tem-se o resultado para as funções com dimensão 100 e as hipóteses aceitas como verdadeiras são das linhas 35 a 45, para o *valor-p* não ajustado; 25 a 45, para o teste Nemenyi, e 24 a 45, para o teste de Shaffer. Portanto, o não ajuste dos valores- p indica que as meta-heurísticas JADE, LSHADE (hipótese da linha 41, *LSHADE vs. JADE*) e PSO (hipótese da linha 38, *PSO vs. JADE*) como as melhores meta-heurísticas (mesma família), e com a aplicação dos procedimentos *post-hoc* SSA (hipótese da linha 28, *SSA vs. JADE*) também possui o melhor desempenho.

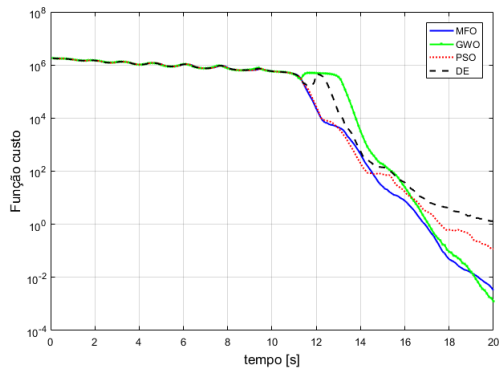
Este trabalho adota a análise feita por (DERRAC et al., 2011) que diz: o teste de Nemenyi além de ter um cálculo mais simples, ele também tem menor poder de minimização do erro FWER (vide seção 2.2.6) do que Shaffer. Assim, as conclusões deste estudo levará em conta os resultados encontrados pelo procedimento de Shaffer.

4.4.1 Análise qualitativa

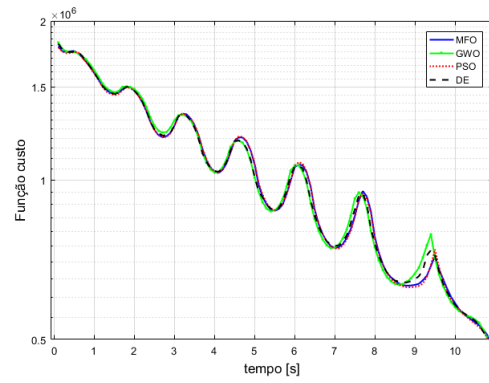
A fim de complementar os resultados, foram feitas análises qualitativas, abordando os estados nos problemas NMPC e o valor das funções da competição CEC2017. Estas análises diferenciam-se da análise quantitativa por mostrarem as demais características dos problemas, além do valor da função custo utilizado para o *ranking* de Friedman, quando utilizadas as meta-heurísticas para resolvê-los.

(a) **Pêndulo:**

O valor da função custo variou ao longo das simulações das meta-heurísticas mais bem ranqueados por Friedman conforme mostrado na Fig. 4.1.



(a) Função custo

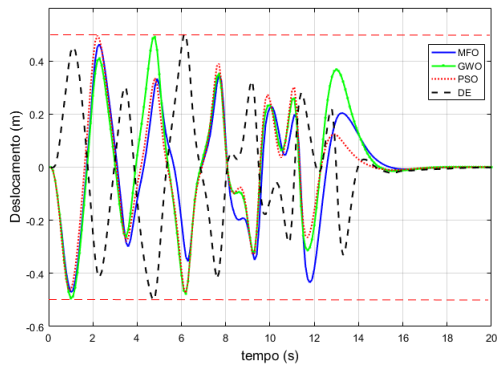


(b) Detalhe entre os instantes 0,1 e 10 segundos

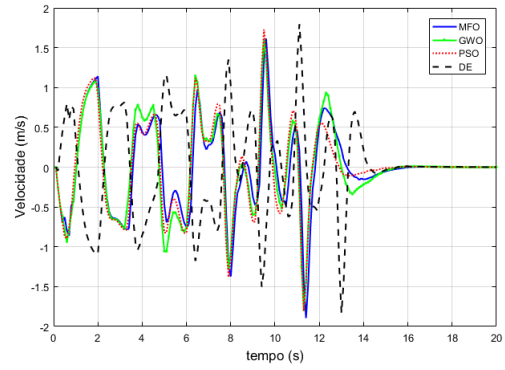
Figura 4.1: Convergência dos valores da função custo - pêndulo

Observa-se o período de oscilação da função custo, primeiros 13 segundos, que é o tempo de inversão do pêndulo (Fig. 4.1(b)). Este instante não é considerado nas comparações já que não reflete o potencial da meta-heurística, pois um modelo bom pode ter variações de custo alta no início, porém podendo controlar o pêndulo invertido em menor tempo.

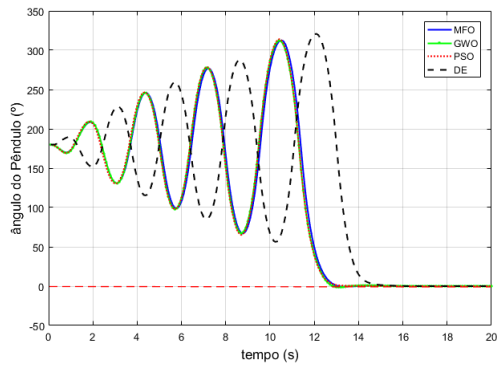
A Fig. 4.1(a) mostra que o GWO possui um custo mais alto entre 12 e 14 segundos, mas após esse tempo converge ao ponto mínimo encontrado entre o grupo. Na Fig. 4.2 é mostrado a variação dos estados ao longo do tempo de simulação.



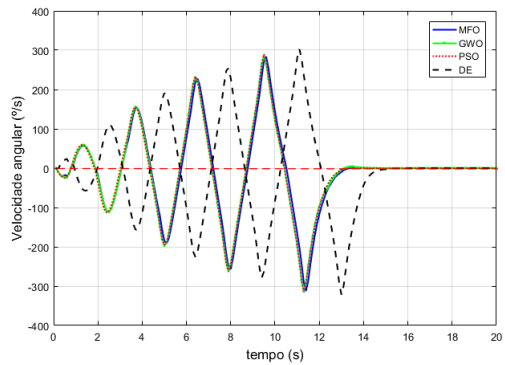
(a) Deslocamento do carro x



(b) Velocidade do carro \dot{x}



(c) Ângulo do pêndulo θ



(d) Velocidade angular do pêndulo $\dot{\theta}$

Figura 4.2: Estados do pêndulo

Nota-se que MFO, PSO e GWO geram semelhante comportamento no pêndulo, já o DE é um pouco mais lento para inverter o pêndulo, indicado na Fig. 4.2.

Adicionalmente, a ação de controle u é mostrada na Fig. 4.3 e o que pode ser inferido é, que apesar do controle ser realizado, há certas variações de comando que não são desejáveis. Entretanto, das variações encontradas, o MFO, o PSO e o GWO resultaram nas menores entre as amostras das meta-heurísticas.

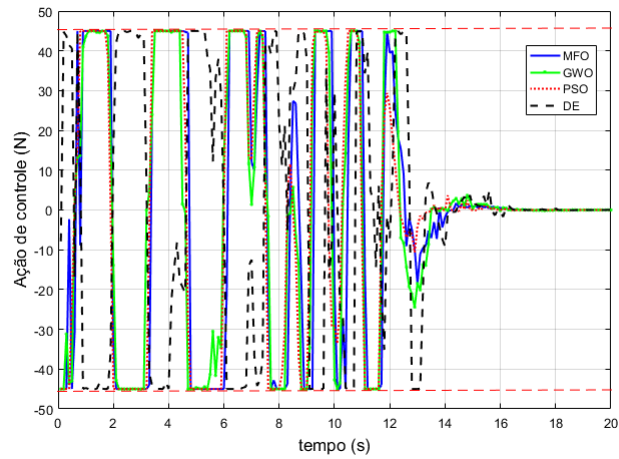
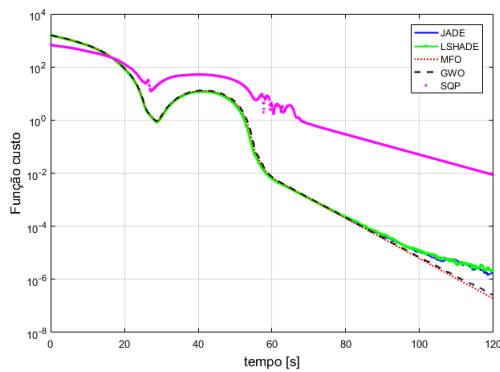


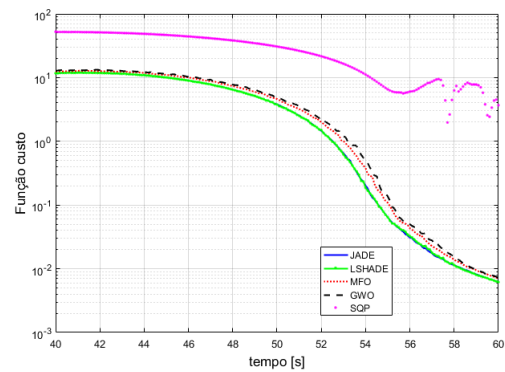
Figura 4.3: Ação de controle u

(b) **Satélite:**

A análise do valor da função custo ao longo do tempo pode ser vista na Fig. 4.4.



(a) Variação do valor na função custo



(b) Detalhe entre o instantes 40 e 60 segundos

Figura 4.4: Convergência dos valores da função custo - satélite

Já na primeira figura (Fig. 4.4(a)) verifica-se a melhora no desempenho com o uso das meta-heurísticas bioinspiradas para o método SQP. A partir de 100 segundos até o final da simulação observa-se que MFO e GWO têm os menores valores. Contudo, como mostrado na Fig.4.4(b) nos demais instantes anteriores o valor para o JADE e LSHADE são ligeiramente menores que eles e é por isso que o *ranking* listou aqueles como os primeiros. Na Fig. 4.5 podem ser vistos os estados e comandos produzidos pelas simulações.

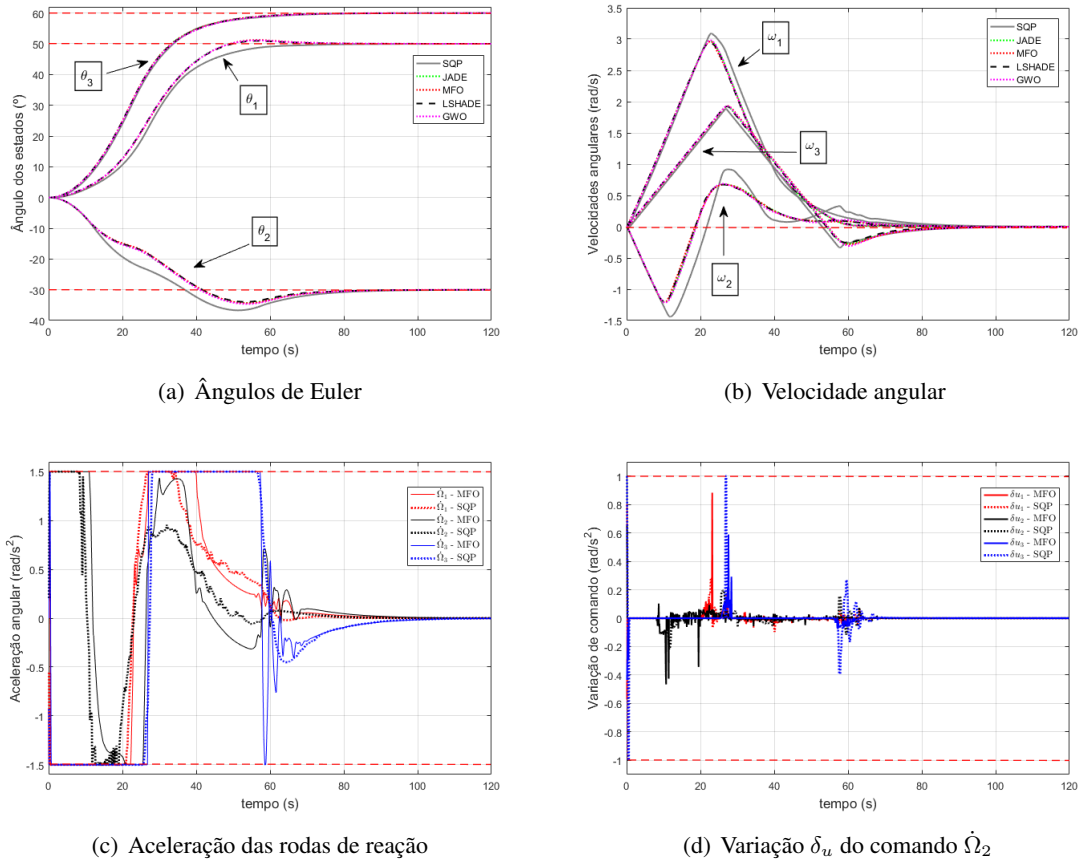


Figura 4.5: Variação nos estados e comandos

Pode-se destacar que o uso dos algoritmos bioinspirados resultaram em valores de estados mais consistentes que o método SQP. Os estados θ_2 , ω_1 e ω_2 tiveram menores variações e amplitudes, observado também pelos valores de *overshoot* dos ângulos de Euler na Tab. 4.11.

Tabela 4.11: *Overshoot* dos ângulos de Euler

Meta-heurística	θ_1 (%)	θ_2 (%)	θ_3 (%)
JADE	2,02	13,59	0,00
MFO	2,37	14,68	0,00
LSHADE	1,96	13,64	0,00
GWO	2,10	14,77	0,00
DA	2,13	13,60	0,00
PSO	2,53	14,95	0,00
DE	2,17	15,39	0,00
SCA	2,05	14,65	0,06
SSA	2,13	15,41	0,00
WOA	1,63	17,63	0,00
SQP	0,01	22,30	0,01

(c) **CEC2017:**

Os valores obtidos pelas meta-heurísticas mais bem colocados no *ranking* da Tab. 4.10 podem ser vistos pela Fig. 4.6.

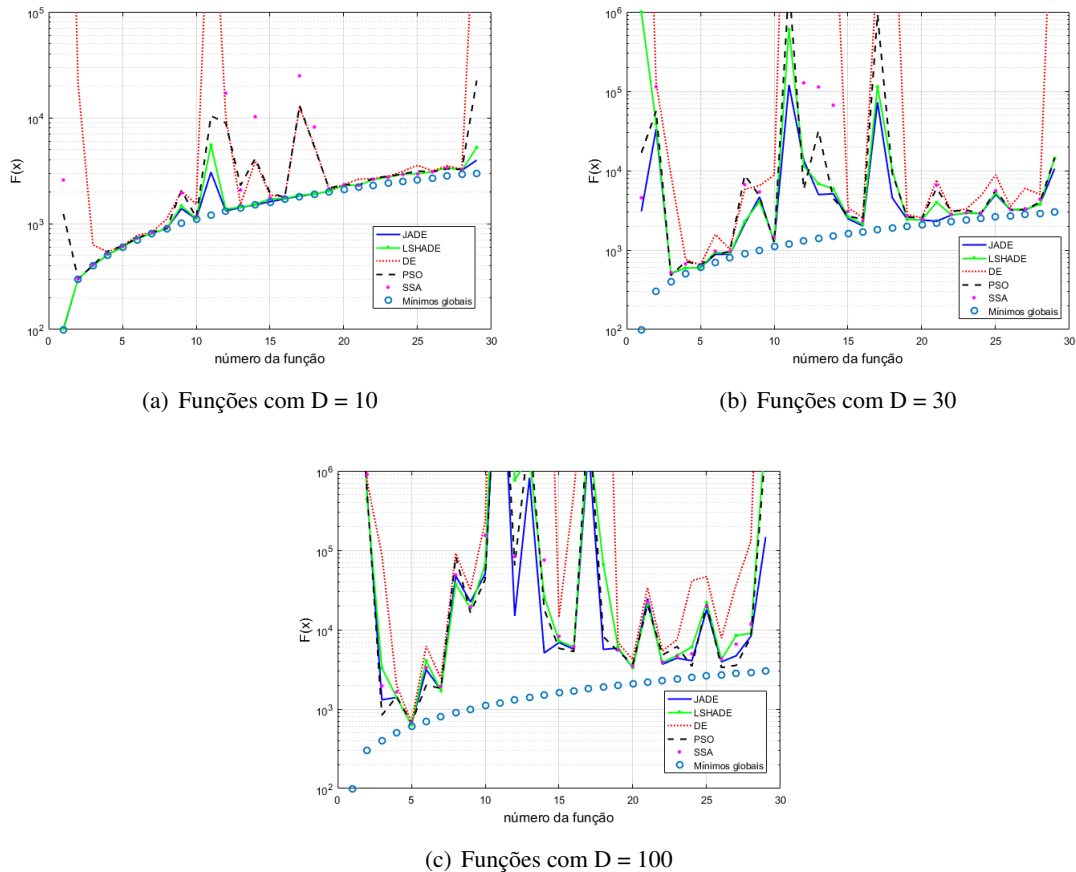


Figura 4.6: Resultados CEC2017 por dimensão

Nos resultados mostrados na Fig. 4.6 é possível observar que, em pequenas dimensões, as meta-heurísticas alcançam a maioria dos mínimos globais. No caso específico do JADE, o mesmo não se aproxima em três funções, especificamente: 9, 11 e 29. Ao aumentar a dimensão o número de funções as quais são atingidas o valor mínimo vai diminuindo, mas é possível observar que o JADE possui o menor valor em funções como a 1, 8, 13, 15, 17, 18, 21 e 29 para $D = 30$ e 12, 14, 18 e 29 para $D = 100$.

4.5 Meta-heurísticas propostas nesta dissertação

Através dos *rankings* de cada problema foram selecionados as melhores meta-heurísticas, dentre os novos, para aplicar as modificações e adaptações. Diversos testes foram realizados e ao final chegou-se em três versões do algoritmo GWO e uma do MFO, as quais podem ser vistas nas subseções a seguir.

4.5.1 GWO - 1ª versão

(a) Descrição

O GWO original mostrado na seção 2.1.4 tem a atualização da posição dos agentes de busca através

da média de três componentes, X_1 , X_2 e X_3 , os quais são calculados de acordo com os três melhores agentes denominados de α , β e δ . Sabe-se que, na natureza, os lobos alfa são os líderes do grupo e sua ordem está acima dos demais. Portanto, aplicar o mesmo peso de importância para as três classes não representaria o real comportamento da alcateia. Por outro lado, de acordo com a ocasião as classes beta e delta podem assumir a voz de comando em caso de alerta de inimigo ou quando encontram a presa antes do alfa.

(b) **Modificações**

Uma nova proposta para a equação (2.31)(c) foi elaborada:

$$X_i^{t+1} = \frac{p_1^{t+1} \cdot X_1 + p_2^{t+1} \cdot X_2 + p_3^{t+1} \cdot X_3}{3} \quad (4.1)$$

onde os valores de p_1 , p_2 e p_3 representam os pesos das classes α , β e δ , respectivamente. Contudo, para esses pesos não podem ser atribuídos valores constantes, já que para determinado momento qualquer uma das classes poderá ter maior importância que as demais. Assim, inspirado nas adaptações dos parâmetros do JADE (vise seção 2.1.3), os pesos da equação (4.1) são adaptativos e seguem a seguinte expressão:

$$p_k^{t+1} = Mp_k^{t+1} + 0.1 \cdot randn(0, 1), \quad k = 1, 2, 3. \quad (4.2)$$

no qual $randn(0, 1)$ é uma função que retorna um valor da distribuição normal padrão e Mp_k é o valor médio de atualização dos pesos e seu valor é obtido por:

$$Mp_k^{t+1} = (1 - c) \cdot Mp_k^t + c \cdot mean(GoodP_k), \quad k = 1, 2, 3. \quad (4.3)$$

onde c é constante de valor 0, 1 e $GoodP_k$ é o arquivo que armazena o valor dos pesos que obtiveram sucesso, ou seja, quando α ($k = 1$), β ($k = 2$) ou δ ($k = 3$) são atualizados. Esse arquivo inicia com 5 valores iguais a 1, ao longo das iterações o arquivo é atualizado retirando o primeiro elemento inserido para sempre permanecer com o mesmo tamanho de 5 componentes.

4.5.2 GWO - 2ª versão

(a) **Descrição**

Analisando os agentes de busca durante a execução das meta-heurísticas (independente do problema), foi observado que próximo ao final das iterações os agentes com melhor solução (α , β e δ) eram diferentes entre si em algumas dimensões, e quando se olha para um problema com dimensão 20 (Pêndulo) essas variações são relevantes na solução.

Para auxiliar na sintonia dos valores de cada dimensão, a 2ª versão do algoritmo GWO foi elaborada e sua inspiração veio da operação de *crossover* do DE.

(b) **Modificações**

Essa versão muda a forma da atualização da partícula com pior solução (X_{last}) durante cada iteração, realizando o processo de cruzamento das informações dos líderes, conforme a expressão seguinte:

$$X_{last,j}^{t+1} = \begin{cases} X_{\alpha,j}^t & se\ r \leq 1 \\ X_{\beta,j}^t & se\ 1 < r \leq 2 \\ X_{\delta,j}^t & se\ 2 < r \leq 3 \end{cases} \quad (4.4)$$

onde r é um valor gerado aleatoriamente no intervalo $[0,3]$ para cada dimensão j .

4.5.3 GWO - 3ª versão

Por se tratar de diferentes operações, as duas versões já mostradas foram incorporadas em uma só, denominada de 3ª versão do GWO.

4.5.4 MFO adaptativo

(a) Descrição

O MFO original mostrado no Capítulo 2 apresenta uma operação para simular o voo de uma mariposa no formato de uma espiral. Essa operação é realizada pela equação (2.32) na qual um dos parâmetros da exponencial é b de valor 2. Testes com outros valores foram realizados e em torno de $b = 1$ obteve-se resultados superiores.

(b) Modificações

Baseado nas técnicas adaptativas do JADE, criou-se a versão do MFO com o parâmetro b adaptativo e individual, ou seja, cada agente de busca possui o valor de b próprio. O cálculo de b_i e de seu valor médio Mb_i para o agente i é idêntico ao mostrado nas equações (4.2) e (4.3) da primeira versão do GWO:

$$\begin{aligned} b_i^{t+1} &= M_b^{t+1} + 0,1 \cdot randn(0,1) \\ Mb_i^{t+1} &= (1 - c) \cdot Mb_i^t + c \cdot mean(GoodB_i) \end{aligned} \quad (4.5)$$

onde c também tem o valor de 0,1. Além dos parâmetros individuais¹ outra variação da versão do GWO criada é o arquivo $GoodB_i$ que, no caso do MFO adaptativo, inicia-se vazio e vai sendo preenchido ao longo das iterações, quando a solução encontrada pelo agente i for melhor que a da iteração anterior, até um tamanho máximo de 5. Ao atingir esse valor o primeiro elemento inserido é retirado para a inserção do novo elemento. O valor de b_i^1 é 1, nas demais iterações, segue o sistema de equações (4.5).

¹No GWO 1ª versão os pesos p_1 , p_2 e p_3 são os mesmos para todos os agentes de busca.

4.6 Desempenho das meta-heurísticas propostas

A mesma metodologia de aquisição dos dados e execução dos estudos de caso foram realizadas com as meta-heurísticas propostas na seção anterior. Houve melhora no desempenho em relação à meta-heurística de origem para os dois problemas de controle implementados. Os resultados podem ser vistos nas subseções a seguir.

4.6.1 Controle do pêndulo invertido simples

Os dados oriundos da execução dos algoritmos GWov1, GWov2, GWov3 e MFOadapt como otimizadores do controle NMPC para o pêndulo podem ser vistos na Tab. A.6. A comparação do desempenho foi feita retirando os quatro piores (JADE, SCA, WOA e SSA) do *ranking* da Tab. 4.9 para manter o número de 10 meta-heurísticas no grupo. Os testes de normalidade e homogeneidade dos dados não foram aplicados já que as outras meta-heurísticas se mostraram não-normais e com diferentes variâncias. O novo *ranking* pode ser visto na Tab. 4.12.

Tabela 4.12: *Ranking* das meta-heurísticas propostas - Pêndulo

Meta-heurística	<i>Ranking</i>
MFOadapt	1,06
GWov1	2,78
GWov3	3,39
MFO	3,81
GWov2	5,09
PSO	5,96
GWO	6,59
DE	8,04
DA	8,45
LSHADE	9,77
χ_f	461,46
valor-p (Friedman)	$1,90 \cdot 10^{-10}$
F_f	316,29
valor-p (IeD)	$4,44 \cdot 10^{-16}$

Observa-se que as três versões do GWO e a versão adaptativa do MFO superaram em *ranking* os originais. Na Tab. 4.13 podem ser vistas as comparações dos pares de meta-heurísticas considerando apenas com as novas versões. Nela verifica-se que apenas p_{Neme} indica semelhança entre o MFOadapt e o GWov1 (hipótese da linha 23). Nos demais valores-p o MFOadapt não possui similaridade a nenhuma outra meta-heurística, o que indica ter o melhor desempenho do grupo. Além disso, para p_{Neme} e p_{Shaf} GWO tem o mesmo desempenho que GWov2 (hipótese da linha 25), indicando que nessa versão não houve melhoria na performance.

Tabela 4.13: Valores- p ajustados - novas versões - Pêndulo

i	Hipótese	valor- p não ajustado	p_{Neme}	p_{Shaf}
1	MFOadap vs .LSHADE	0	0	0
2	MFOadap vs .DA	0	0	0
3	MFOadap vs .DE	0	0	0
4	GWOv1 vs .LSHADE	0	0	0
5	GWOv3 vs .LSHADE	0	0	0
6	GWOv1 vs .DA	0	0	0
7	MFOadap vs .GWO	0	0	0
8	GWOv1 vs .DE	0	0	0
9	GWOv3 vs .DA	0	0	0
10	MFOadap vs .PSO	0	0	0
11	GWOv2 vs .LSHADE	0	0	0
12	GWOv3 vs .DE	0	0	0
13	GWOv2 vs .MFOadap	0	0	0
14	GWOv1 vs .GWO	0	0	0
15	GWOv2 vs .DA	0	0	0
16	GWOv3 vs .GWO	0	0	0
17	GWOv1 vs .PSO	0	0	0
18	GWOv2 vs .DE	0	0	0
19	MFOadap vs .MFO	0	0	0
20	GWOv3 vs .PSO	0	0,0001	0
21	GWOv3 vs .MFOadap	0	0,0009	0,0003
22	GWOv1 vs .GWOv2	0	0,0011	0,0004
23	GWOv1 vs .MFOadap	0,0016	0,0760	0,0219
24	GWOv2 vs .GWOv3	0,0018	0,0842	0,0219
25	GWOv2 vs .GWO	0,0065	0,2927	0,0650
26	GWOv2 vs .MFO	0,0196	0,8855	0,1377
27	GWOv1 vs .MFO	0,0595	1,0	0,3574
28	GWOv2 vs .PSO	0,1129	1,0	0,5649
29	GWOv1 vs .GWOv3	0,2685	1,0	1,0
30	GWOv3 vs .MFO	0,4368	1,0	1,0

As melhorias demonstradas pelos testes estatísticos podem ser observadas pela convergência dos valores da função custo na Fig. 4.7.

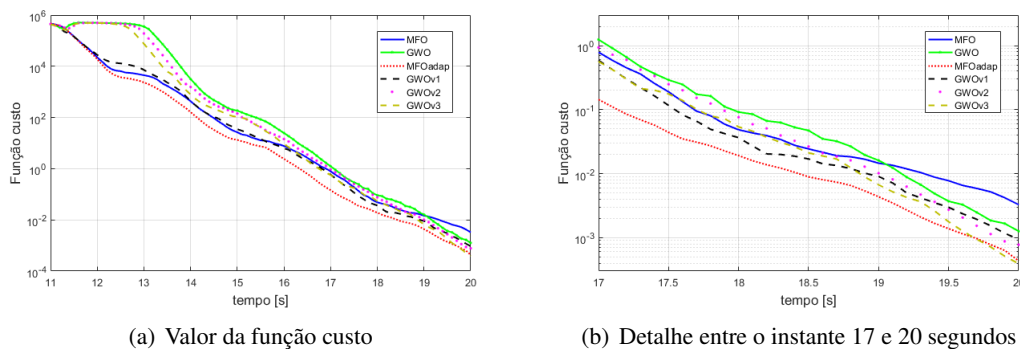


Figura 4.7: Convergência das meta-heurísticas propostas - pêndulo

É possível observar que o MFOadap é superior às outras meta-heurísticas em praticamente todo o tempo de simulação, já que no último segundo o GWOv3 consegue menor valor da função. De um modo geral, todas as versões apresentaram menor valor de convergência do que as meta-heurísticas originais.

4.6.2 Controle de atitude de satélites

Os dados da simulação das novas versões como otimizadoras do NMPC para o controle de satélite podem ser vistos na Tab. A.7. Da mesma forma que a subseção anterior, foram retirados os quatro últimas meta-heurísticas do *ranking* (SQP, WOA, SSA e SCA) para a inserção das novas versões. O novo *ranking* é mostrado na Tab 4.14.

Tabela 4.14: *Ranking* das meta-heurísticas propostas - Satélite

Meta-heurística	<i>Ranking</i>
GWOv3	2,75
GWOv2	3,31
JADE	4,72
MFOadap	5,26
LSHADE	5,06
MFO	6,10
DA	6,43
GWOv1	7,11
GWO	7,69
PSO	8,67
DE	8,85
χ_f	227,92
valor-p (Friedman)	$1,38 \cdot 10^{-10}$
F_f	35,46
valor-p (IeD)	$1,38 \cdot 10^{-54}$

Verifica-se, que as modificações no GWO e no MFO os superaram em *ranking* também para o problema de controle do satélite. Os APVs podem ser vistos na Tab. 4.15. Os três valores-p indicados nela mostram que MFOadap tem, estatisticamente, o mesmo desempenho do MFO (hipótese 29), assim como, GWOv1 com o GWO. Ou seja, as novas versões não se mostraram superiores aos originais no controle do satélite. Além disso, a hipótese 32 também é aceita como verdadeira para todos os valores-p o que indica que GWOv3 e GWOv2 obtiveram o mesmo desempenho. Assim, a adição do GWOv1 no GWOv2 não tornou este melhor.

Como a hipótese 20 (GWOv2 vs JADE) é rejeitada por todos os testes (valores- $p \leq \alpha$), assim, conclui-se que as técnicas implementadas no GWOv2 demonstraram ser superiores a todas as meta-heurísticas experimentadas.

Tabela 4.15: Valores- p ajustados - novas versões - Satélite

i	Hipótese	valor- p não ajustado	p_{Neme}	p_{Shaf}
1	GWov3 vs .DE	0	0	0
2	GWov3 vs .PSO	0	0	0
3	GWov2 vs .DE	0	0	0
4	GWov2 vs .PSO	0	0	0
5	GWov3 vs .GWO	0	0	0
6	GWov2 vs .GWO	0	0	0
7	GWov1 vs .GWov3	0	0	0
8	GWov1 vs .GWov2	0	0	0
9	GWov3 vs .DA	0	0	0
10	MFOadap vs .DE	0	0	0
11	MFOadap vs .PSO	0	0	0
12	GWov3 vs .MFO	0	0	0
13	GWov2 vs .DA	0	0	0
14	GWov2 vs .MFO	0	0,0001	0,0001
15	GWov3 vs .MFOadap	0	0,0013	0,0007
16	MFOadap vs .GWO	0	0,0025	0,0014
17	GWov1 vs .JADE	0	0,0033	0,0017
18	GWov3 vs .LSHADE	0,0001	0,0056	0,0029
19	GWov1 vs .LSHADE	0,0005	0,0321	0,0157
20	GWov3 vs .JADE	0,0009	0,0500	0,0227
21	GWov2 vs .MFOadap	0,0010	0,0578	0,0262
22	GWov1 vs .MFOadap	0,0019	0,1063	0,0464
23	GWov2 vs .LSHADE	0,0033	0,1818	0,0760
24	GWov1 vs .DE	0,0034	0,1899	0,0760
25	GWov1 vs .PSO	0,0086	0,4745	0,1639
26	GWov2 vs .JADE	0,0178	0,9804	0,3208
27	MFOadap vs .DA	0,0496	1,0	0,6949
28	GWov1 vs .MFO	0,0905	1,0	0,9965
29	MFOadap vs .MFO	0,1591	1,0	1,0
30	GWov1 vs .DA	0,2554	1,0	1,0
31	GWov1 vs .GWO	0,3296	1,0	1,0
32	GWov2 vs .GWov3	0,3432	1,0	1,0
33	MFOadap vs .JADE	0,3643	1,0	1,0
34	MFOadap vs .LSHADE	0,7350	1,0	1,0

Na Fig. 4.8 é observada convergência das meta-heurísticas propostas. A partir de 60 segundos GWov2 e GWov3 se destacam por resultarem em menores valores da função custo que os demais (Fig. 4.8(a)). Em outros instantes, também é possível observar (Fig. 4.8(b)) que essas duas meta-heurísticas possuem melhor convergência do resultado.

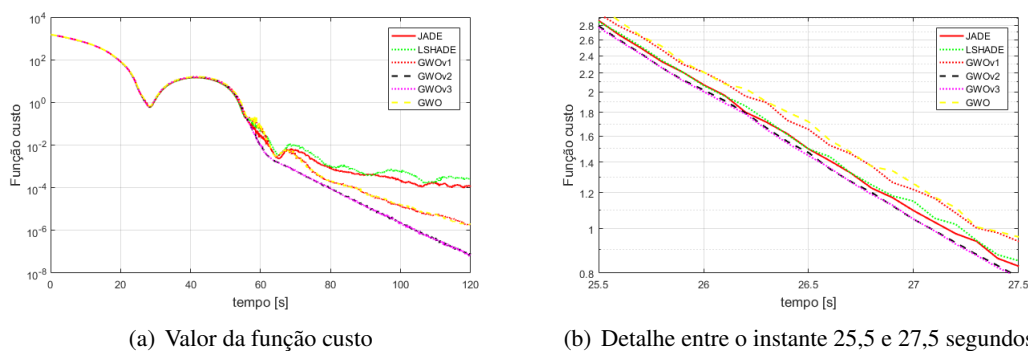
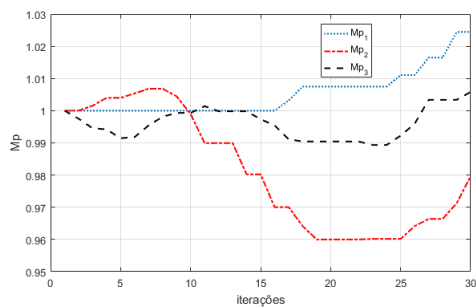


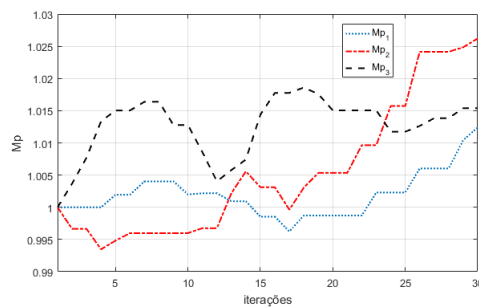
Figura 4.8: Convergência das meta-heurísticas propostas - satélite

4.6.3 Validação das adaptações de GWov1 e MFOadap

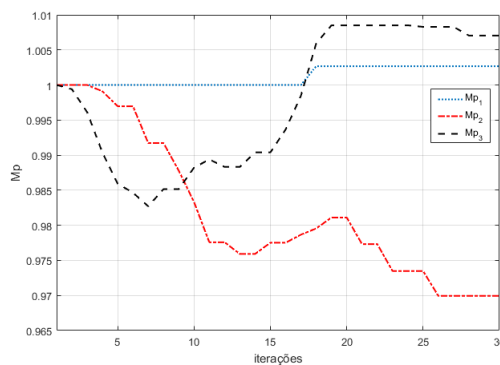
A vantagem de aplicar a técnica de adaptação nos parâmetros de uma meta-heurística é que seus valores vão sendo direcionados de acordo com o estado que o problema for adquirindo. Na Fig. 4.9, por exemplo, pode ser observado três instantes em que o valor de Mp_k variam para valores diferentes. Esses valores são a referência para os parâmetros p_1 , p_2 e p_3 da versão GWov1.



(a) Instante 0,2 segundo



(b) Instante 0,5 segundo



(c) Instante 14 segundos

Figura 4.9: Variação de Mp - pêndulo

Da mesma forma, o Mb do MFOadap guia o parâmetro b e cada problema exige valores diferentes, conforme mostrado na Fig. 4.10.

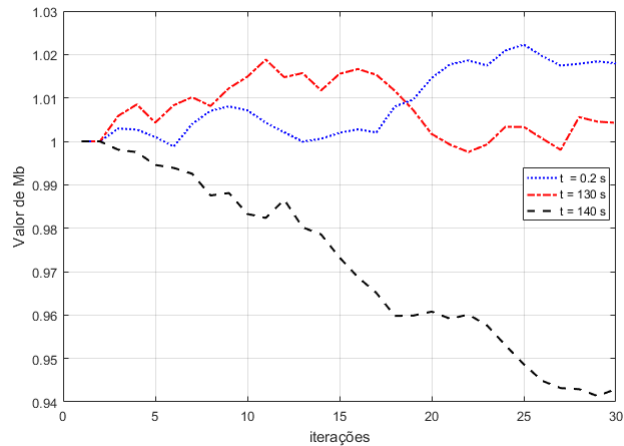


Figura 4.10: Variação dos valores de Mb em diferentes instantes

Por esses resultados verifica-se a importância da implementação das adaptações dos parâmetros, pois, problemas como o de controle do pêndulo exige uma sintonia diferente a cada instante de amostragem. Contudo, não são todos os problemas que necessitam desses métodos. Neste sentido, as funções da competição CEC2017 e o controle do satélite não tiveram melhor performance com as versões adaptativas do GWO e MFO.

4.7 Análise do tempo de execução

Os tempos de execução foram extraídos pela função '*tic; toc;*' do MatLab e as especificações do ambiente de teste são as mostradas pela Tab. 4.16.

Tabela 4.16: Especificações da máquina utilizada nas simulações

Sistema Operacional	Windows 10 Home
Processador	Intel(R) Core(TM) i3 CPU M 370
Velocidade do Processador	2,40 GHz
Memória instalada	4GB
Software	MatLab R2016a

Assim, na tabelas 4.17 a 4.19 podem ser vistas a análise do tempo de execução de cada meta-heurística segundo a metodologia aplicada.

Tabela 4.17: Tempo de execução das meta-heurísticas - pêndulo

Meta-heurística	\bar{T}_1	$max(T_1)$	$min(T_1)$	\bar{T}_1/T_0
MFO	0,254(T_0)	0,271	0,249	1,000
PSO	0,253	0,267	0,250	0,996
GWO	0,254	0,275	0,251	1,003
DE	0,764	0,828	0,749	3,009
DA	0,356	0,521	0,328	1,402
LSHADE	0,214	0,260	0,207	0,842
JADE	0,271	0,339	0,265	1,070
SCA	0,255	0,270	0,252	1,004
WOA	0,253	0,275	0,247	0,999
SSA	0,259	0,293	0,255	1,021
MFOadapt	0,267	0,293	0,263	1,054
GWOv1	0,255	0,289	0,249	1,003

Tabela 4.18: Tempo de execução das meta-heurísticas - satélite

Meta-heurística	\bar{T}_1	$max(T_1)$	$min(T_1)$	\bar{T}_1/T_0
JADE	0,020(T_0)	0,092	0,018	1,000
LSHADE	0,024	0,088	0,022	1,247
MFO	0,013	0,035	0,012	0,658
GWO	0,012	0,035	0,011	0,621
DA	0,021	0,077	0,016	1,072
DE	0,035	0,054	0,034	1,763
SCA	0,012	0,032	0,011	0,629
PSO	0,012	0,026	0,011	0,606
SSA	0,013	0,038	0,012	0,632
WOA	0,012	0,037	0,011	0,629
GWOv3	0,030	0,067	0,028	1,49
GWOv2	0,029	0,057	0,028	1,46

Tabela 4.19: Tempo de execução das meta-heurísticas - CEC2017

D=10			D=30			D=100		
Meta-heurística	\bar{T}_1	\bar{T}_1/T_0	Meta-heurística	\bar{T}_1	\bar{T}_1/T_0	Meta-heurística	\bar{T}_1	\bar{T}_1/T_0
JADE	0,61(T_0)	1,00	JADE	0,73(T_0)	1,00	JADE	1,76(T_0)	1,00
LSHADE	0,70	1,16	LSHADE	0,77	1,06	LSHADE	1,40	0,79
DE	0,74	1,23	DE	1,01	1,38	PSO	1,35	0,77
PSO	0,21	0,34	PSO	0,33	0,45	SSA	1,62	0,92
SSA	0,50	0,82	SSA	0,63	0,87	DE	3,56	2,02
GWO	0,32	0,54	GWO	0,49	0,67	GWO	1,62	0,92
MFO	0,36	0,59	MFO	0,51	0,70	MFO	1,62	0,92
SCA	0,31	0,51	SCA	0,46	0,63	WOA	1,42	0,81
WOA	0,30	0,50	WOA	0,43	0,59	DA	31,56	17,93
DA	16,62	27,25	DA	28,84	39,64	SCA	1,58	0,90

Para o problema do pêndulo pode ser observado que grande parte dos experimentos obtiveram tempos iguais (em média) ao MFO (melhor desempenho de controle), lembrando que neste problema o número de iterações é de 30. A meta-heurística DE e DA se mostraram 3 e 1,4 vezes, respectivamente, mais demorado que o MFO. Já o LSHADE foi 16% mais rápido. As versões criadas MFOadapt e GWOv1 se mostraram com tempos de execução semelhantes ao do MFO. Mas, para um sistema em tempo real com $\tau_s = 0,1s$ nenhuma meta-heurística estaria apta a ser utilizada.

No problema do satélite (30 iterações) tem-se que o DE possui o pior tempo de execução (76% maior) e o PSO, o melhor (39% menor). O LSHADE não se mostrou mais rápido pois a sua operação de diminuir o número de agentes ao longo das iterações possui um mínimo de 4, o qual é justamente o tamanho da população da simulação, resultando assim em 25% mais lento que o JADE. Além dele, as versões GWOv3

e GWOv2 são 50% mais lentas que o JADE e levam para executar o problema 2 vezes mais tempo que o GWO original. Contudo, verificando a média, o menor e o maior valor atingidos pelas meta-heurísticas, todos poderiam ser aplicados em tempo real, já que nenhum deles passou o tempo de amostragem de 0,1s.

O PSO foi a meta-heurística com o menor tempo de execução na resolução da função 18 do *benchmark* CEC2017 (1000 iterações) nas três dimensões analisadas. A meta-heurística DA mostrou-se bastante morosa, chegando a ser 40 vezes mais lento que o JADE na dimensão 30. Isso se deve ao fato de o DA realizar cálculos de diferença e soma entre os agentes para cada dimensão. Assim, o tempo de execução do DA aumenta quanto maior for o número de iterações e dimensões do problema.

Capítulo 5

Conclusões

5.1 Conclusões do Trabalho

O presente trabalho contribui por apresentar novas versões de alguns dos mais recentes algoritmos bioinspirados encontrados na literatura, a fim de proporcionar desempenho superior aos originais. Essas versões incluem modificações e adaptações em seus parâmetros. A comparação de desempenho é feita por testes não-paramétricos para múltiplos dados (NxN meta-heurísticas). Três estudos de casos são usados neste trabalho: dois otimizadores do controle preditivo não-linear (pêndulo e simulação de satélites) e um *benchmark* da competição CEC2017 com funções mono-objetivas com restrição de fronteira.

O primeiro problema de controle tem o objetivo de inverter um pêndulo simples e mantê-lo invertido na posição de referência ($x = 0$). Já o segundo, refere-se ao controle de atitude de satélites cujo modelo é obtido de uma plataforma de testes desenvolvido por Gonzales (2009).

Através do teste de Friedman (FRIEDMAN, 1937) obteve-se o ranking da Tab. 4.9, posteriormente, são aplicados dois procedimentos post-hoc, Nemenyi (NEMENYI, 1963) e Shaffer (SHAFFER, 1986), para análise em pares de meta-heurísticas (1x1), considerando os valores encontrados por Friedman. Entretanto, de acordo com Derrac et al. (2011) Shaffer possui mais poder de conclusão que o Nemenyi. Diante disso, os resultados encontrados pelo procedimento de Shaffer foram considerados.

Conclusão 1: O resultado apresentado foi que, estatisticamente, a meta-heurística MFO (mariposas) teve o melhor desempenho para o pêndulo e JADE. **Conclusão 2:** MFO e LSHADE foram os melhores para o controle de atitude de satélites. Além disso, a utilização dos algoritmos bioinspirados resultou em uma melhor performance que o método analítico SQP aplicado por Rodrigues (2018).

Conclusão 3: As funções do *benchmark* CEC2017 foram executadas em três dimensões, 10, 30 e 100. De acordo com os procedimentos *post-hoc* e o *ranking* da Tab. 4.10, para $D = 10$, o DE e suas versões JADE e LSHADE tiveram os melhores desempenhos. Para $D = 30$, o resultado foi o JADE, o LSHADE e o PSO, quando aumenta a dimensão para 100 adiciona-se a essa lista o SSA (salpas).

Baseadas nas comparações, foram apresentadas duas versões auto-adaptativas do GWO e MFO, inspiradas nas técnicas encontradas no JADE, denominadas GWOv1 e MFOadapt. Uma modificação do GWO, baseada na etapa de *crossover* do DE, foi titulada de GWOv2. E por último a junção das versões GWOv1

e GWOv2 resultou no GWOv3.

Conclusão 4: Pode-se destacar que a contribuição deste trabalho foi apresentar meta-heurísticas com modificações simples que se mostraram superiores às originais para os estudos de caso de controle preditivo. O MFOadapt teve o melhor desempenho para o problema do pêndulo e, GWOv3 e GWOv2, para o controle de atitude de satélites.

Conclusão 5: Os desempenhos das adaptações e modificações não foram mais satisfatórios nas funções da competição pois, as meta-heurísticas JADE e LSHADE foram desenvolvidas tendo como base funções similares às do *benchmark*, ou seja, foram testadas para o melhor desempenho possível nestes tipos de problemas. Assim, este trabalho reforça a afirmação do teorema da inexistência de almoço grátis (WOLPERT; MACREADY et al., 1995), o qual diz que se uma meta-heurística A é melhor que B para um grupo de n problemas, existirá outro grupo de n problemas nos quais B será melhor que A.

5.2 Sugestões para Trabalhos Futuros

As contribuições futuras que podem ser feitas a este trabalho são as seguintes:

- Realizar a implementação dos problemas de controle não-linear baseado nas meta-heurísticas em tempo real, a fim de validar as simulações aqui apresentadas;
- Testar as novas meta-heurísticas implementadas neste trabalho em outros estudos de caso, verificando em mais problemas de controle a eficiência aqui já demonstrada;
- Realizar um estudo mais detalhado do tempo de execução dos algoritmos, verificando a capacidade de paralelização das meta-heurísticas em linguagem específica de *hardware*.
- Utilizar as meta-heurísticas como sintonizadoras dos parâmetros de controle do NMPC, referentes à parametrização e ao sinal filtrado;

REFERÊNCIAS BIBLIOGRÁFICAS

ALAMIR, M. *A pragmatic story of model predictive control: self-contained algorithms and case-studies*. [S.l.]: CreateSpace Independent Publishing Platform, 2013.

ALANIZ, A. *Model predictive control with application to real-time hardware and guided parafoil*. Tese (Doutorado) — Massachusetts Institute of Technology, 2004.

ALCALÁ-FDEZ, J. et al. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, Citeseer, v. 17, 2011.

ALCALÁ-FDEZ, J. et al. Keel: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, Springer, v. 13, n. 3, p. 307–318, 2009.

ALI, E. Automatic tuning of model predictive controllers based on fuzzy logic. *optimization*, v. 2, n. 2, p. 1, 2001.

ARSHAM, H. Kuiper's p-value as a measuring tool and decision procedure for the goodness-of-fit test. *Journal of Applied Statistics*, Taylor & Francis, v. 15, n. 2, p. 131–135, 1988.

AWAD, N. et al. Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective bound constrained real-parameter numerical optimization. In: *Technical Report*. [S.l.]: NTU, Singapore, 2016.

BINITHA, S.; SATHYA, S. S. et al. A survey of bio inspired optimization algorithms. *International Journal of Soft Computing and Engineering*, v. 2, n. 2, p. 137–151, 2012.

CAMACHO, E. F.; BORDONS, C. A. *Model predictive control in the process industry*. [S.l.]: Springer Science & Business Media, 2012.

CARVALHO, T. C. d. S. de. *Probabilidade e estatística*. [S.l.: s.n.], 2015.

CONOVER, W. J. *Practical nonparametric statistics*. Wiley New York, 1999.

CORREIA, M. S. B. B. *Probabilidade e estatística*. [S.l.: s.n.], 2003.

CUI, L. et al. Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations. *Computers & Operations Research*, Elsevier, v. 67, p. 155–173, 2016.

D'AGOSTINO, R.; PEARSON, E. S. Tests for departure from normality. empirical results for the distributions of b_2 and b . *Biometrika*, Oxford University Press, v. 60, n. 3, p. 613–622, 1973.

DALLAL, G. E.; WILKINSON, L. An analytic approximation to the distribution of lilliefors's test statistic for normality. *The American Statistician*, Taylor & Francis Group, v. 40, n. 4, p. 294–296, 1986.

DAS, S.; SUGANTHAN, P. N. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation*, IEEE, v. 15, n. 1, p. 4–31, 2011.

- DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, v. 7, n. Jan, p. 1–30, 2006.
- DERRAC, J. et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, Elsevier, v. 1, n. 1, p. 3–18, 2011.
- DEVORE, J. L. *PROBABILIDADE E ESTATÍSTICA PARA ENGENHARIA E CIÊNCIAS*. [S.l.]: Cengage Learning Edições Ltda., 2010.
- EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. In: IEEE. *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*. [S.l.], 1995. p. 39–43.
- FEIJÓ, B.; SILVA, F. S. C. da; CLUA, E. *Introdução à Ciência da Computação com Jogos*. [S.l.]: Elsevier, 2009.
- FISHER, R. A. *Statistical methods and scientific inference*. Oxford, England: Hafner Publishing Co., 1956.
- FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937.
- GARCÍA, S. et al. A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. *Soft Computing*, Springer, v. 13, n. 10, p. 959, 2009.
- GARCIA, S.; HERRERA, F. An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *Journal of Machine Learning Research*, v. 9, n. Dec, p. 2677–2694, 2008.
- GARCÍA, S. et al. A study on the use of non-parametric tests for analyzing the evolutionary algorithms’ behaviour: a case study on the cec’2005 special session on real parameter optimization. *Journal of Heuristics*, Springer, v. 15, n. 6, p. 617, 2009.
- GONZALES, R. G. *Utilização dos métodos SDRE e Filtro de Kalman para o controle de atitude de simuladores de satélites*. Tese (Doutorado) — Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espacial, DEM/INPE, São Jose dos Campos, SP, 2009.
- HARTLEY, E. N. et al. Predictive control of a boeing 747 aircraft using an fpga. 2012.
- HARTLEY, E. N. et al. Predictive control using an fpga with application to aircraft control. *IEEE Transactions on Control Systems Technology*, IEEE, v. 22, n. 3, p. 1006–1017, 2014.
- HEPPNER, F.; GREANDER, U. A stochastic nonlinear model for coordinated bird flocks. *The ubiquity of chaos*, Washington: AAAS, p. 233–238, 1990.
- HOLLAND, J. H. Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, SIAM, v. 2, n. 2, p. 88–105, 1973.
- IMAN, R. L.; DAVENPORT, J. M. Approximations of the critical region of the fbietkan statistic. *Communications in Statistics-Theory and Methods*, Taylor & Francis, v. 9, n. 6, p. 571–595, 1980.
- JOHNSON, N. L.; KEMP, A. W.; KOTZ, S. *Univariate discrete distributions*. [S.l.]: John Wiley & Sons, 2005. v. 444.

- KAWAI, F. et al. Automatic tuning for model predictive control: Can particle swarm optimization find a better parameter? In: IEEE. *Intelligent Control, 2007. ISIC 2007. IEEE 22nd International Symposium on*. [S.l.], 2007. p. 646–651.
- LEVENE, H. Robust tests for equality of variances. *Contributions to probability and statistics. Essays in honor of Harold Hotelling*, Stanford University Press, p. 279–292, 1961.
- LILLIEFORS, H. W. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American statistical Association*, Taylor & Francis Group, v. 62, n. 318, p. 399–402, 1967.
- LING, K.-V.; WU, B. F.; MACIEJOWSKI, J. Embedded model predictive control (mpc) using a fpga. *IFAC Proceedings Volumes*, Elsevier, v. 41, n. 2, p. 15250–15255, 2008.
- LIU, J.; LAMPINEN, J. A fuzzy adaptive differential evolution algorithm. *Soft Computing*, Springer, v. 9, n. 6, p. 448–462, 2005.
- MACIEJOWSKI, J. M. *Predictive control: with constraints*. [S.l.]: Pearson education, 2002.
- MERCIECA, J.; FABRI, S. G. Particle swarm optimization for nonlinear model predictive control. *Proc. ADVCOMP*, p. 88–93, 2011.
- MERCIECA, J.; FABRI, S. G. A metaheuristic particle swarm optimization approach to nonlinear model predictive control. *International Journal On Advances in Intelligent Systems*, Citeseer, v. 5, n. 3, 2012.
- MESQUITA, E. M.; QUINTERO, C. H. L.; SAMPAIO, R. C. Bio-inspired optimization applied to the tuning of model predictive control parameters. In: 24TH ABCM INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING. [S.l.], 2017.
- MIRJALILI, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, Elsevier, v. 89, p. 228–249, 2015.
- MIRJALILI, S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, Springer, v. 27, n. 4, p. 1053–1073, 2016.
- MIRJALILI, S. Sca: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, Elsevier, v. 96, p. 120–133, 2016.
- MIRJALILI, S. et al. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, Elsevier, v. 114, p. 163–191, 2017.
- MIRJALILI, S.; LEWIS, A. The whale optimization algorithm. *Advances in Engineering Software*, Elsevier, v. 95, p. 51–67, 2016.
- MIRJALILI, S.; MIRJALILI, S. M.; LEWIS, A. Grey wolf optimizer. *Advances in engineering software*, Elsevier, v. 69, p. 46–61, 2014.
- MONTGOMERY, D. C.; RUNGER, G. C. *Applied statistics and probability for engineers*. [S.l.]: John Wiley & Sons, 2010.
- MURILO, A.; ALAMIR, M.; ALBERER, D. A general nmpe framework for a diesel engine air path. *International Journal of Control*, Taylor & Francis, v. 87, n. 10, p. 2194–2207, 2014.
- NEMENYI, P. Distribution-free multiple comparisons (doctoral dissertation, princeton university, 1963). *Dissertation Abstracts International*, v. 25, n. 2, p. 1233, 1963.

- ĐŽOLMOGOROV, . Sulla determinazione empirica di una leggi di distribuzione. *Giorn. Ist it lit Đ; Ital. Attuari*, 4 (1933), 83, v. 91, 1933.
- OMRAN, M. G.; SALMAN, A.; ENGELBRECHT, A. P. Self-adaptive differential evolution. In: SPRINGER. *International Conference on Computational and Information Science*. [S.l.], 2005. p. 192–199.
- ÖNER, M.; KOCAKOÇ, İ. D. Jmasm 49: A compilation of some popular goodness of fit tests for normal distribution: Their algorithms and matlab codes (matlab). *Journal of Modern Applied Statistical Methods*, v. 16, n. 2, p. 30, 2017.
- POLI, R.; KENNEDY, J.; BLACKWELL, T. Particle swarm optimization. *Swarm intelligence*, Springer, v. 1, n. 1, p. 33–57, 2007.
- QIN, A. K.; HUANG, V. L.; SUGANTHAN, P. N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE transactions on Evolutionary Computation*, IEEE, v. 13, n. 2, p. 398–417, 2009.
- RAWLINGS, J. B.; MAYNE, D. Q. *Model predictive control: Theory and design*. [S.l.]: Nob Hill Pub., 2009.
- RIBOLDI, J. et al. Precisão e poder de testes de homocedasticidade paramétricos e não-paramétricos avaliados por simulação. *Revista Brasileira de Biomedicina*, v. 32, n. 3, p. 334–344, 2014.
- RODRIGUES, R. S. *Desenvolvimento de controlador preditivo para controle de atitude de satélites e validação em HIL*. Dissertação (Mestrado) — Dissertação de Mestrado do Curso de Pós-Graduação em Sistemas Mecatrônicos, PPMEC-Universidade de Brasília, Brasília, DF, 2018.
- RODRIGUES, R. S.; MURILO, A.; SOUZA, L. C. G. de. Projeto de controlador preditivo baseado em modelo para sistemas de controle de atitude de satélites artificiais. *Revista Interdisciplinar de Pesquisa em Engenharia-RIPE*, v. 2, n. 20, p. 179–194, 2017.
- ROYSTON, J. An extension of shapiro and wilk's w test for normality to large samples. *Applied statistics*, JSTOR, p. 115–124, 1982.
- SAMPAIO, R. C. et al. Novel architectures for particle swarm optimization implementation on hardware applied to nonlinear model predictive control. 2018.
- SANDERS, M. *Characteristic function of the central chi-squared distribution*. [S.l.], 2009.
- SHAFFER, J. P. Modified sequentially rejective multiple test procedures. *Journal of the American Statistical Association*, Taylor & Francis Group, v. 81, n. 395, p. 826–831, 1986.
- SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). *Biometrika*, JSTOR, v. 52, n. 3/4, p. 591–611, 1965.
- SHI, Y.; EBERHART, R. A modified particle swarm optimizer. In: IEEE. *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. [S.l.], 1998. p. 69–73.
- SIEGEL, S.; JR, N. J. C. *Estatística não-paramétrica para ciências do comportamento*. [S.l.]: Artmed Editora, 1975.
- SIVANANAITHAPERUMAL, S. et al. Pso algorithm based nonlinear model predictive control. In: *Proceedings of the international conference on advances in control and optimization of dynamic systems, ACODS 2007, February*. [S.l.: s.n.], 2007. p. 1–2.

- SMIRNOV, N. Table for estimating the goodness of fit of empirical distributions. *The annals of mathematical statistics*, JSTOR, v. 19, n. 2, p. 279–281, 1948.
- SOONG, T. *Modelos probabilísticos em engenharia e ciências*. [S.l.]: LTC, 1986.
- STORN, R.; PRICE, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, Springer, v. 11, n. 4, p. 341–359, 1997.
- TANABE, R.; FUKUNAGA, A. Success-history based parameter adaptation for differential evolution. In: IEEE. *Evolutionary Computation (CEC), 2013 IEEE Congress on*. [S.l.], 2013. p. 71–78.
- TANABE, R.; FUKUNAGA, A. S. Improving the search performance of shade using linear population size reduction. In: IEEE. *Evolutionary Computation (CEC), 2014 IEEE Congress on*. [S.l.], 2014. p. 1658–1665.
- TRAWIŃSKI, B. et al. Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *International Journal of Applied Mathematics and Computer Science*, Versita, v. 22, n. 4, p. 867–881, 2012.
- TRIGUERO, I. et al. Keel 3.0: an open source software for multi-stage analysis in data mining. *International Journal of Computational Intelligence Systems*, Atlantis Press, v. 10, n. 1, p. 1238–1249, 2017.
- WANG, L. *Model predictive control system design and implementation using MATLAB®*. [S.l.]: Springer Science & Business Media, 2009.
- WEISE, T. Global optimization algorithms-theory and application. *Self-published*, v. 2, 2009.
- WOLPERT, D. H.; MACREARY, W. G. et al. *No free lunch theorems for search*. [S.l.], 1995.
- YAMASHITA, A.; ZANIN, A.; ODLOAK, D. Tuning of model predictive control with multi-objective optimization. *Brazilian Journal of Chemical Engineering*, SciELO Brasil, v. 33, n. 2, p. 333–346, 2016.
- YANG, X.-S. *Nature-inspired algorithms and applied optimization*. [S.l.]: Springer, 2017. v. 744.
- ZHANG, J.; SANDERSON, A. C. Jade: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, IEEE, v. 13, n. 5, p. 945–958, 2009.

APÊNDICES

A. RESULTADO MEDIANA 51 SIMULAÇÕES

Tabela A.1: Dados das simulações para o Pêndulo

Problemas	PSO	DE	DA	SSA	WOA	MFO	GWO	SCA	JADE	LSHADE
1	196.96	560.27	3631.89	1015730.54	573704.97	430.79	3167.14	459489.74	1433.50	1231.84
2	137.30	373.40	2479.84	950516.28	575498.04	301.71	2022.86	422621.02	1222.09	967.73
3	112.90	304.31	1713.42	919027.32	583325.83	213.22	1279.02	373191.23	1072.15	968.23
4	85.76	279.57	1216.93	919324.28	582118.00	162.37	847.48	352735.74	1067.95	797.75
5	82.91	234.58	937.09	911597.01	578469.21	123.61	621.95	289585.89	904.15	615.42
6	84.34	204.08	797.83	903022.08	565591.36	94.33	472.20	187399.66	885.49	594.91
7	77.09	170.51	623.60	894681.99	563841.16	69.53	379.56	121851.58	772.84	583.08
8	79.60	171.69	508.16	912599.92	557149.72	53.07	294.22	92895.97	822.71	599.01
9	81.78	146.84	404.67	916202.88	549083.02	39.27	232.53	60541.11	796.79	611.50
10	80.71	142.25	344.70	962032.74	543718.15	31.41	198.97	43737.56	858.26	623.16
11	76.54	139.70	306.49	974915.10	533003.43	26.45	184.91	26291.20	931.13	604.80
12	71.14	136.00	253.89	1021134.05	521804.60	21.58	157.09	15646.33	961.98	601.72
13	71.26	116.41	234.02	1026500.83	514591.33	20.22	132.89	9430.11	956.95	660.36
14	71.80	105.68	197.89	1013816.68	503523.12	16.64	113.55	6230.19	890.05	648.04
15	57.60	91.74	171.30	978895.49	505647.59	14.70	99.42	4596.58	949.31	615.32
16	41.50	75.72	146.68	952613.65	495517.23	12.80	84.26	3554.08	987.88	623.02
17	30.46	62.09	116.22	898671.25	496637.65	12.27	70.79	2961.79	968.06	619.58
18	27.09	48.13	101.81	887240.23	497888.12	11.40	56.41	2530.71	1037.78	528.45
19	23.24	45.92	82.61	861416.29	471178.89	9.70	42.72	2302.34	790.50	521.36
20	20.40	41.06	56.73	866071.83	488995.33	8.88	32.27	2240.42	859.86	396.89
21	16.74	37.65	33.49	865373.95	481921.15	7.67	24.82	1635.97	660.99	371.03
22	13.06	29.25	26.08	856242.44	444545.28	6.41	18.66	1546.58	704.73	346.67
23	11.55	24.94	27.85	853087.31	400660.72	5.16	13.99	1510.92	671.52	315.17
24	10.55	21.28	24.83	874189.11	326283.02	4.22	9.93	1461.40	613.79	286.05
25	9.05	18.78	12.99	924599.93	259383.34	3.43	7.63	1472.86	561.99	244.78
26	7.65	15.89	10.00	927086.29	356654.57	2.66	5.89	1335.72	496.60	195.11
27	6.35	12.19	9.85	970167.51	284948.27	2.03	4.49	1332.92	512.05	166.00
28	5.03	10.99	6.68	996743.49	187252.03	1.59	3.23	1227.35	474.17	137.31
29	3.87	10.08	6.10	952661.64	155000.28	1.21	2.30	1047.45	541.64	133.56
30	3.32	9.04	5.31	914520.80	112705.10	1.01	1.67	871.04	519.34	101.00
31	3.01	6.99	5.04	905820.33	87913.31	0.79	1.24	883.82	525.74	92.64
32	2.72	6.42	4.19	867313.30	72954.33	0.59	0.91	743.07	486.62	87.86
33	2.37	6.30	3.90	836463.96	62414.54	0.45	0.66	679.49	574.11	77.53
34	2.00	5.75	3.35	830374.37	55286.72	0.36	0.47	661.77	419.16	76.93
35	1.52	5.40	3.07	816868.60	35736.23	0.26	0.36	604.09	424.68	71.73
36	1.25	4.86	2.55	815687.38	36371.73	0.19	0.29	613.52	442.77	74.55
37	1.00	4.56	1.92	814320.61	34814.31	0.13	0.25	600.99	435.59	75.09
38	0.83	4.63	2.00	821968.31	34992.36	0.09	0.17	598.91	327.05	70.47
39	0.68	4.40	1.53	841292.25	30202.84	0.08	0.16	491.44	353.66	63.26
40	0.59	4.32	1.53	871894.69	26026.08	0.06	0.11	440.68	313.95	69.48
41	0.62	4.08	1.31	908227.09	23595.20	0.05	0.09	395.22	377.11	66.29
42	0.62	3.74	1.20	877836.40	20153.12	0.04	0.08	344.91	420.09	57.90
43	0.55	3.14	1.07	907645.13	18544.99	0.04	0.07	258.13	307.65	48.14
44	0.59	3.03	0.97	917705.80	16173.38	0.03	0.06	198.83	369.12	55.64
45	0.63	2.98	1.06	901452.26	13804.40	0.03	0.05	203.61	304.56	52.24
46	0.57	2.81	0.77	858021.24	16333.95	0.02	0.05	150.99	305.74	49.96
47	0.56	2.70	0.66	835664.13	18428.98	0.02	0.03	152.81	273.79	49.76
48	0.51	2.59	0.52	807851.26	17044.41	0.02	0.03	151.56	246.32	49.12
49	0.48	2.14	0.46	781163.16	15522.58	0.02	0.03	139.11	267.87	49.41
50	0.43	2.03	0.48	764841.82	15119.19	0.02	0.02	111.15	257.64	41.74
51	0.35	2.28	0.48	768013.84	14245.22	0.01	0.02	97.75	300.41	40.60
52	0.26	1.76	0.37	767748.20	14789.21	0.01	0.01	87.03	360.55	40.32
53	0.23	1.70	0.34	776479.69	14472.80	0.01	0.01	75.60	346.59	41.19
54	0.23	1.73	0.29	779644.28	13398.00	0.01	0.01	71.63	295.49	40.65
55	0.21	1.59	0.32	808870.65	13644.51	0.01	0.00	58.98	286.11	39.53
56	0.18	1.62	0.29	828824.09	14535.10	0.01	0.00	57.93	265.88	38.47
57	0.17	1.60	0.23	836053.73	14014.69	0.01	0.00	58.52	274.82	42.39
58	0.17	1.47	0.23	869501.00	13678.39	0.01	0.00	54.14	270.15	38.03
59	0.14	1.36	0.23	881156.11	11802.73	0.01	0.00	51.80	264.18	36.40
60	0.12	1.31	0.24	863937.29	11846.31	0.00	0.00	45.46	247.44	32.69
61	0.12	1.33	0.23	844465.68	11522.16	0.00	0.00	40.04	221.54	30.09

Tabela A.2: Dados das simulações para o Satélite

Problemas	PSO	DE	DA	SSA	WOA	MFO	GWO	SCA	JADE	LSHADE	SQP
1	1454.35	1448.64	1428.47	1470.54	1445.21	1431.60	1437.93	1435.45	1428.06	1427.52	648.07
2	1298.03	1294.80	1280.14	1304.07	1291.69	1282.26	1286.73	1285.22	1279.78	1279.34	603.15
3	1149.24	1145.93	1131.75	1155.04	1141.92	1133.74	1137.93	1136.83	1131.14	1130.70	562.51
4	1002.87	1000.49	986.66	1009.33	994.95	988.38	992.25	990.91	986.06	985.62	518.56
5	864.07	861.60	848.96	869.63	856.53	851.21	854.85	853.55	849.05	848.61	471.67
6	736.34	734.38	723.30	740.81	728.13	725.95	728.31	727.06	723.36	722.96	422.64
7	616.94	615.07	604.47	621.11	608.80	607.34	609.05	607.23	604.55	604.13	376.52
8	502.97	501.78	490.93	506.79	495.40	493.91	495.62	493.87	491.09	490.79	337.24
9	395.68	394.37	384.46	399.74	388.94	387.68	388.75	387.09	384.53	384.29	298.68
10	297.28	296.58	287.20	300.85	291.74	290.55	290.76	289.59	287.18	286.97	258.58
11	210.68	210.59	202.08	215.24	206.50	205.58	205.33	204.29	202.02	201.79	216.41
12	139.09	139.84	132.16	143.99	136.55	135.47	135.07	134.17	132.11	131.90	172.15
13	85.08	86.77	79.94	91.10	84.93	83.50	82.51	82.11	80.14	79.88	127.42
14	47.77	49.82	44.28	54.05	49.21	47.13	46.25	46.87	44.26	44.23	90.82
15	21.76	23.13	19.41	25.79	22.58	21.41	20.70	21.15	19.43	19.39	57.02
16	6.59	7.19	5.62	8.48	6.78	6.40	6.15	6.27	5.60	5.62	30.80
17	1.80	2.00	1.66	2.46	2.05	1.79	1.84	1.91	1.64	1.68	23.13
18	0.79	0.90	0.73	1.38	1.37	0.77	0.82	0.95	0.70	0.71	15.97
19	1.26	1.21	1.20	1.26	1.65	1.19	1.28	1.35	1.15	1.17	24.91
20	3.08	2.98	2.92	3.00	3.60	2.96	3.09	3.19	2.84	2.87	34.01
21	5.64	5.57	5.34	5.49	6.48	5.51	5.78	5.82	5.22	5.25	41.30
22	8.52	8.47	7.90	8.38	9.72	8.29	8.59	8.46	7.83	7.88	46.14
23	11.20	11.23	10.31	11.28	12.82	10.99	11.04	11.33	10.28	10.26	48.57
24	13.49	13.57	12.37	13.64	15.52	13.18	13.31	13.41	12.26	12.31	50.52
25	15.19	15.25	13.75	15.53	17.29	14.81	14.80	15.11	13.67	13.78	51.78
26	16.17	16.36	14.45	16.72	18.57	15.85	15.75	15.79	14.45	14.38	51.88
27	16.30	16.55	14.47	17.16	18.88	15.91	15.83	15.86	14.37	14.32	50.77
28	15.70	16.02	13.84	16.64	18.19	15.19	15.13	15.02	13.60	13.53	48.48
29	14.32	14.69	12.32	15.42	17.01	13.83	13.60	13.67	12.14	12.08	45.07
30	12.34	12.70	10.31	13.39	15.09	11.83	11.50	11.62	10.09	9.99	40.63
31	9.87	10.33	7.92	10.96	12.36	9.36	8.99	9.14	7.68	7.57	35.21
32	7.14	7.61	5.44	8.14	9.31	6.68	6.34	6.52	5.20	5.09	28.77
33	4.44	4.97	3.07	5.34	6.33	4.08	3.80	3.87	2.88	2.81	21.32
34	2.16	2.56	1.27	2.82	3.46	1.95	1.74	1.80	1.13	1.08	13.05
35	0.66	0.86	0.31	1.04	1.37	0.57	0.49	0.50	0.28	0.27	6.52
36	0.18	0.22	0.16	0.27	0.33	0.17	0.18	0.30	0.13	0.16	7.61
37	0.09	0.20	0.11	0.42	0.30	0.10	0.16	0.51	0.07	0.10	6.55
38	0.04	0.16	0.06	0.62	0.44	0.06	0.08	0.54	0.03	0.05	3.10
39	0.02	0.10	0.03	0.38	0.48	0.02	0.03	0.36	0.01	0.03	3.43
40	0.01	0.03	0.01	0.19	0.24	0.01	0.01	0.17	0.01	0.01	2.54
41	0.00	0.01	0.00	0.05	0.08	0.00	0.00	0.07	0.00	0.00	3.39
42	0.00	0.01	0.01	0.02	0.02	0.00	0.01	0.05	0.00	0.01	1.62
43	0.00	0.01	0.01	0.03	0.02	0.00	0.01	0.06	0.01	0.01	0.96
44	0.00	0.01	0.01	0.04	0.05	0.00	0.00	0.09	0.01	0.01	0.75
45	0.00	0.01	0.01	0.04	0.06	0.00	0.00	0.10	0.00	0.01	0.63
46	0.00	0.00	0.01	0.02	0.05	0.00	0.00	0.10	0.00	0.01	0.54
47	0.00	0.00	0.01	0.01	0.03	0.00	0.00	0.07	0.00	0.00	0.47
48	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.03	0.00	0.00	0.41
49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.35
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.31
51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.27
52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.23
53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.20
54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.18
55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.15
56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.13
57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.12
58	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.10
59	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.09
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.08
61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.07
62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.06

Tabela A.3: Dados das simulações para as funções do CEC2017 - $D = 10$, ($\cdot 10^3$)

Problemas	PSO	DE	DA	SSA	WOA	MFO	GWO	SCA	JADE	LSHADE
1	1.29	795.83	354155.64	2.70	18100.47	7.07	135.62	985819.88	0.10	0.10
2	0.30	1.05	16.02	0.30	4.07	11.16	1.78	2.02	0.30	0.30
3	0.40	0.41	0.50	0.41	0.43	0.41	0.42	0.45	0.40	0.40
4	0.54	0.51	0.56	0.52	0.56	0.53	0.52	0.55	0.50	0.51
5	0.60	0.60	0.63	0.61	0.64	0.60	0.60	0.62	0.60	0.60
6	0.72	0.72	0.76	0.73	0.79	0.74	0.73	0.78	0.72	0.72
7	0.82	0.81	0.85	0.82	0.84	0.83	0.82	0.85	0.81	0.81
8	0.90	0.90	1.82	0.94	1.48	1.16	0.91	1.04	0.90	0.90
9	1.87	1.74	2.44	1.93	2.27	2.07	1.70	2.46	1.25	1.36
10	1.13	1.11	1.42	1.17	1.20	1.21	1.13	1.22	1.10	1.10
11	12.20	55.90	8853.86	1677.43	2847.07	111.93	566.18	17052.55	1.56	1.75
12	8.18	1.34	17.61	14.00	18.57	9.38	10.96	49.86	1.31	1.31
13	1.89	1.42	3.50	1.55	2.22	3.08	3.74	1.71	1.40	1.42
14	2.22	1.52	33.66	4.27	6.32	5.01	5.25	2.82	1.50	1.50
15	1.86	1.62	2.00	1.75	2.00	1.76	1.75	1.79	1.60	1.60
16	1.75	1.72	1.85	1.79	1.79	1.77	1.76	1.80	1.70	1.70
17	9.32	1.91	35.69	19.15	18.19	22.36	34.84	236.43	1.82	1.82
18	4.20	1.90	34.77	3.48	33.17	12.01	11.32	4.65	1.90	1.90
19	2.14	2.02	2.15	2.09	2.20	2.14	2.10	2.10	2.00	2.00
20	2.33	2.31	2.35	2.31	2.35	2.33	2.32	2.22	2.31	2.31
21	2.30	2.30	2.36	2.30	2.32	2.31	2.31	2.39	2.30	2.30
22	2.67	2.61	2.67	2.62	2.65	2.63	2.62	2.66	2.61	2.61
23	2.79	2.74	2.80	2.75	2.79	2.77	2.76	2.79	2.74	2.74
24	2.94	2.94	3.01	2.94	2.95	2.95	2.95	2.97	2.94	2.94
25	2.90	3.00	3.68	2.90	3.55	3.03	2.98	3.10	2.90	2.90
26	3.08	3.10	3.14	3.09	3.14	3.10	3.10	3.10	3.09	3.09
27	3.28	3.41	3.46	3.22	3.48	3.41	3.41	3.27	3.38	3.38
28	3.26	3.16	3.32	3.22	3.39	3.25	3.20	3.24	3.16	3.17
29	16.56	27.83	2114.65	423.77	1012.12	678.38	456.43	1293.23	3.55	3.59

Tabela A.4: Dados das simulações para as funções do CEC2017 - $D = 30$, ($\cdot 10^4$)

Problemas	PSO	DE	DA	SSA	WOA	MFO	GWO	SCA	JADE	LSHADE
1	0.31	105154.31	1183737.99	0.22	309562.64	1085899.15	315195.89	1995305.77	0.01	0.23
2	2.27	4.15	23.38	5.62	25.01	18.64	5.75	7.51	0.82	1.09
3	0.05	0.06	0.21	0.05	0.11	0.13	0.06	0.28	0.05	0.05
4	0.07	0.06	0.08	0.07	0.08	0.07	0.06	0.08	0.06	0.06
5	0.06	0.06	0.07	0.07	0.07	0.06	0.06	0.07	0.06	0.06
6	0.08	0.09	0.13	0.09	0.13	0.13	0.09	0.12	0.08	0.08
7	0.09	0.09	0.11	0.10	0.11	0.10	0.09	0.11	0.09	0.09
8	0.55	0.15	1.16	0.49	1.00	0.73	0.25	0.79	0.09	0.14
9	0.45	0.83	0.80	0.52	0.76	0.55	0.47	0.87	0.43	0.38
10	0.12	0.15	0.81	0.14	0.77	0.39	0.21	0.36	0.12	0.13
11	69.69	717.53	73729.19	1259.27	33627.30	7257.98	8102.37	234423.20	3.44	6.37
12	1.15	2.75	14973.17	7.93	354.79	445.06	16.81	97373.78	0.51	0.78
13	1.34	0.93	90.85	4.67	139.91	26.20	16.52	52.68	0.16	0.16
14	0.40	1.14	63.59	5.49	221.24	4.31	9.43	4145.33	0.18	0.19
15	0.25	0.23	0.43	0.30	0.43	0.33	0.27	0.41	0.24	0.23
16	0.23	0.19	0.28	0.23	0.28	0.25	0.21	0.28	0.20	0.20
17	50.12	16.00	1024.01	74.43	785.22	149.84	92.10	988.74	2.64	2.93
18	0.56	0.73	6099.66	316.34	752.74	16.80	96.28	8478.62	0.21	0.21
19	0.26	0.23	0.30	0.26	0.29	0.28	0.25	0.29	0.23	0.23
20	0.25	0.24	0.26	0.25	0.27	0.25	0.24	0.26	0.24	0.24
21	0.60	0.93	0.98	0.62	0.88	0.71	0.59	1.01	0.23	0.23
22	0.30	0.28	0.32	0.28	0.31	0.29	0.28	0.31	0.27	0.27
23	0.31	0.29	0.34	0.29	0.32	0.30	0.30	0.33	0.29	0.29
24	0.29	0.30	0.36	0.29	0.32	0.33	0.30	0.35	0.29	0.29
25	0.56	0.49	0.87	0.53	0.86	0.60	0.50	0.77	0.43	0.42
26	0.32	0.33	0.34	0.33	0.35	0.33	0.33	0.36	0.32	0.32
27	0.32	0.34	0.39	0.33	0.37	0.44	0.35	0.43	0.32	0.32
28	0.39	0.37	0.53	0.43	0.54	0.43	0.40	0.51	0.36	0.36
29	0.99	2.83	7493.11	807.59	4209.98	34.08	1103.84	16617.12	0.84	1.03

Tabela A.5: Dados das simulações para as funções do CEC2017 - $D = 100$, ($\cdot 10^5$)

Problemas	PSO	DE	DA	SSA	WOA	MFO	GWO	SCA	JADE	LSHADE
1	775.12	532375.03	1667668.41	6505.40	910170.87	1415498.30	657484.05	2070448.31	22.05	36617.03
2	4.43	6.86	14.14	7.03	9.07	9.95	5.02	5.16	4.71	3.23
3	0.01	0.05	0.38	0.01	0.16	0.23	0.07	0.47	0.01	0.01
4	0.01	0.01	0.02	0.01	0.02	0.02	0.01	0.02	0.01	0.01
5	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
6	0.02	0.03	0.04	0.03	0.04	0.06	0.02	0.04	0.02	0.03
7	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.01
8	0.72	0.52	1.02	0.36	0.75	0.54	0.52	0.91	0.24	0.25
9	0.16	0.33	0.31	0.18	0.29	0.18	0.18	0.33	0.22	0.18
10	0.18	0.50	4.16	0.75	2.57	2.14	0.86	1.57	0.14	0.18
11	1038.61	91128.64	512023.19	9721.15	211406.93	445518.79	136458.50	984559.28	124.01	1210.35
12	0.28	6762.91	62594.46	0.87	15951.32	60485.17	15656.09	162024.46	0.08	0.24
13	24.07	31.14	490.94	58.87	185.15	119.26	87.74	505.70	1.43	4.31
14	0.04	1007.13	20652.31	0.90	2450.84	21496.53	1641.96	54715.92	0.04	0.07
15	0.05	0.06	0.14	0.08	0.16	0.08	0.07	0.15	0.07	0.06
16	0.05	0.05	0.31	0.06	0.15	0.09	0.06	0.45	0.05	0.05
17	48.55	30.22	845.46	71.86	152.15	172.02	86.97	983.33	4.46	9.42
18	0.04	675.95	23522.02	233.49	3245.23	8331.51	2704.60	50994.68	0.04	0.05
19	0.05	0.07	0.08	0.06	0.07	0.06	0.05	0.08	0.06	0.06
20	0.03	0.03	0.04	0.03	0.04	0.04	0.03	0.04	0.03	0.03
21	0.19	0.35	0.33	0.20	0.31	0.21	0.21	0.35	0.25	0.21
22	0.05	0.04	0.06	0.04	0.05	0.04	0.04	0.05	0.03	0.03
23	0.05	0.05	0.09	0.04	0.07	0.05	0.05	0.07	0.04	0.04
24	0.03	0.07	0.19	0.04	0.10	0.15	0.07	0.21	0.03	0.04
25	0.18	0.19	0.43	0.18	0.37	0.22	0.19	0.39	0.14	0.15
26	0.03	0.04	0.07	0.04	0.06	0.04	0.04	0.09	0.04	0.04
27	0.03	0.12	0.24	0.04	0.13	0.20	0.10	0.27	0.04	0.05
28	0.07	0.08	0.25	0.11	0.18	0.11	0.09	0.29	0.07	0.08
29	5.47	4006.03	70853.52	2576.28	23184.54	29912.75	13262.98	127677.33	0.17	2.92

Tabela A.6: Dados das simulações das modificações e adaptações propostas ao GWO e MFO aplicadas ao problema do pêndulo

Problemas	GWOv1	GWOv2	GWOv3	MFOadap
1	435.61	1550.17	795.54	160.86
2	304.34	1160.94	539.92	113.13
3	218.70	793.88	399.30	80.73
4	162.33	551.90	314.86	56.32
5	127.35	410.17	257.92	42.03
6	107.16	321.67	218.04	32.82
7	89.79	264.99	187.01	25.55
8	67.23	233.86	156.51	20.36
9	51.90	194.81	128.30	16.50
10	43.70	168.42	113.14	14.36
11	34.87	142.65	102.85	13.32
12	30.40	101.89	98.70	11.72
13	25.09	84.44	86.22	10.61
14	20.98	67.21	63.51	8.95
15	17.31	52.37	51.52	8.14
16	14.50	42.94	39.95	7.41
17	11.88	32.35	29.68	6.93
18	9.64	25.32	23.12	5.35
19	8.73	21.30	17.57	4.11
20	7.50	17.83	12.46	3.10
21	6.21	14.44	9.21	2.43
22	5.46	11.47	6.95	1.84
23	4.09	8.27	4.94	1.44
24	3.23	6.09	3.96	1.10
25	2.50	4.57	2.89	0.83
26	2.04	3.56	2.10	0.63
27	1.68	2.58	1.44	0.46
28	1.37	2.07	1.14	0.33
29	1.04	1.65	0.87	0.24
30	0.79	1.29	0.70	0.19
31	0.59	0.93	0.56	0.14
32	0.42	0.72	0.42	0.11
33	0.30	0.60	0.30	0.08
34	0.22	0.42	0.21	0.07
35	0.16	0.33	0.20	0.06
36	0.12	0.25	0.18	0.04
37	0.08	0.20	0.14	0.03
38	0.07	0.15	0.10	0.03
39	0.05	0.12	0.08	0.03
40	0.04	0.10	0.07	0.02
41	0.04	0.08	0.05	0.02
42	0.03	0.06	0.05	0.02
43	0.02	0.05	0.04	0.01
44	0.02	0.04	0.03	0.01
45	0.02	0.03	0.03	0.01
46	0.02	0.03	0.02	0.01
47	0.01	0.02	0.02	0.01
48	0.01	0.02	0.02	0.01
49	0.01	0.02	0.01	0.01
50	0.01	0.01	0.01	0.01
51	0.01	0.01	0.01	0.00
52	0.01	0.01	0.01	0.00
53	0.01	0.01	0.00	0.00
54	0.00	0.00	0.00	0.00
55	0.00	0.00	0.00	0.00
56	0.00	0.00	0.00	0.00
57	0.00	0.00	0.00	0.00
58	0.00	0.00	0.00	0.00
59	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00
61	0.00	0.00	0.00	0.00

Tabela A.7: Dados das simulações das modificações e adaptações propostas ao GWO e MFO aplicadas ao problema do satélite

Problemas	GWOv1	GWOv2	GWOv3	MFOadap
1	1435.79	1429.08	1428.94	1430.02
2	1285.89	1280.73	1280.58	1281.78
3	1136.50	1132.09	1131.73	1133.10
4	991.18	986.65	986.49	987.62
5	853.34	849.58	849.46	850.36
6	727.10	723.66	723.50	724.72
7	607.94	604.64	604.60	606.68
8	494.33	491.14	491.14	493.79
9	387.52	384.50	384.51	387.18
10	289.86	287.07	287.08	289.97
11	204.33	201.88	201.78	204.93
12	133.95	131.75	131.63	134.99
13	81.66	79.56	79.44	82.86
14	45.71	43.92	43.75	46.88
15	20.34	19.13	19.11	21.25
16	5.99	5.47	5.44	6.32
17	1.81	1.59	1.61	1.78
18	0.80	0.72	0.70	0.78
19	1.30	1.30	1.30	1.18
20	3.18	3.14	3.13	2.91
21	5.78	5.68	5.67	5.40
22	8.62	8.41	8.37	8.14
23	11.03	10.97	10.92	10.80
24	13.21	13.06	12.99	13.09
25	14.78	14.57	14.46	14.67
26	15.63	15.35	15.25	15.58
27	15.70	15.31	15.22	15.66
28	14.96	14.55	14.45	15.09
29	13.47	13.03	12.94	13.65
30	11.38	10.92	10.88	11.64
31	8.90	8.42	8.40	9.25
32	6.25	5.81	5.82	6.56
33	3.72	3.34	3.34	4.00
34	1.65	1.40	1.40	1.87
35	0.44	0.36	0.35	0.54
36	0.18	0.11	0.11	0.16
37	0.15	0.04	0.04	0.10
38	0.08	0.01	0.01	0.05
39	0.03	0.00	0.00	0.02
40	0.01	0.00	0.00	0.01
41	0.00	0.00	0.00	0.00
42	0.00	0.00	0.00	0.00
43	0.01	0.00	0.00	0.00
44	0.00	0.00	0.00	0.00
45	0.00	0.00	0.00	0.00
46	0.00	0.00	0.00	0.00
47	0.00	0.00	0.00	0.00
48	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	0.00
50	0.00	0.00	0.00	0.00
51	0.00	0.00	0.00	0.00
52	0.00	0.00	0.00	0.00
53	0.00	0.00	0.00	0.00
54	0.00	0.00	0.00	0.00
55	0.00	0.00	0.00	0.00
56	0.00	0.00	0.00	0.00
57	0.00	0.00	0.00	0.00
58	0.00	0.00	0.00	0.00
59	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00
61	0.00	0.00	0.00	0.00
62	0.00	0.00	0.00	0.00

B. RESULTADO DOS PROCEDIMENTOS POST-HOC

A aplicação dos procedimentos post-hoc resultou em valores-p ajustados (APVs) das comparações em pares do ranking de Friedman. Esses resultados podem ser vistos nas tabelas de B.1 a B.5 para os estudos de caso analisados neste trabalho.

Tabela B.1: Valores-p ajustados - Pêndulo

i	Hipótese	valor-p não ajustado	p_{Neme}	p_{Shaf}
1	SSA vs .MFO	0	0	0
2	WOA vs .MFO	0	0	0
3	PSO vs .SSA	0	0	0
4	SSA vs .GWO	0	0	0
5	PSO vs .WOA	0	0	0
6	MFO vs .SCA	0	0	0
7	WOA vs .GWO	0	0	0
8	MFO vs .JADE	0	0	0
9	DE vs .SSA	0	0	0
10	DA vs .SSA	0	0	0
11	PSO vs .SCA	0	0	0
12	GWO vs .SCA	0	0	0
13	PSO vs .JADE	0	0	0
14	DE vs .WOA	0	0	0
15	GWO vs .JADE	0	0	0
16	MFO vs .LSHADE	0	0	0
17	DA vs .WOA	0	0	0
18	SSA vs .LSHADE	0	0	0
19	DE vs .SCA	0	0	0
20	DA vs .MFO	0	0	0
21	PSO vs .LSHADE	0	0	0
22	WOA vs .LSHADE	0	0	0
23	GWO vs .LSHADE	0	0	0
24	DA vs .SCA	0	0.000001	0
25	DE vs .MFO	0	0.000001	0
26	DE vs .JADE	0	0.000009	0.000004
27	SSA vs .JADE	0	0.000014	0.000006
28	DA vs .JADE	0.000001	0.000057	0.000023
29	SSA vs .SCA	0.000022	0.000978	0.00037
30	PSO vs .DA	0.000264	0.011886	0.004226
31	DA vs .GWO	0.000523	0.02353	0.007843
32	SCA vs .LSHADE	0.000652	0.029347	0.00913
33	PSO vs .DE	0.001004	0.045192	0.013056
34	WOA vs .JADE	0.001004	0.045192	0.013056
35	DE vs .GWO	0.001871	0.084215	0.020586
36	DE vs .LSHADE	0.007782	0.350211	0.077825
37	JADE vs .LSHADE	0.01103	0.496342	0.099268
38	MFO vs .GWO	0.01201	0.540451	0.099268
39	WOA vs .SCA	0.015429	0.694322	0.108006
40	PSO vs .MFO	0.019678	0.885509	0.118068
41	DA vs .LSHADE	0.021305	0.958741	0.118068
42	SSA vs .WOA	0.06814	1.0	0.272559
43	SCA vs .JADE	0.38584	1.0	1.0
44	DE vs .DA	0.719718	1.0	1.0
45	PSO vs .GWO	0.85761	1.0	1.0

Tabela B.2: Valores- p ajustados - Satélite

i	Hipótese	valor- p não ajustado	p_{Neme}	p_{Shaf}
1	JADE vs .SQP	0	0	0
2	MFO vs .SQP	0	0	0
3	LSHADE vs .SQP	0	0	0
4	WOA vs .JADE	0	0	0
5	WOA vs .MFO	0	0	0
6	WOA vs .LSHADE	0	0	0
7	SSA vs .JADE	0	0	0
8	GWO vs .SQP	0	0	0
9	SSA vs .MFO	0	0	0
10	DA vs .SQP	0	0	0
11	SSA vs .LSHADE	0	0	0
12	SCA vs .JADE	0	0	0
13	MFO vs .SCA	0	0	0
14	SCA vs .LSHADE	0	0	0
15	WOA vs .GWO	0	0	0
16	DA vs .WOA	0	0	0
17	SSA vs .GWO	0	0.000001	0.000001
18	PSO vs .SQP	0	0.000002	0.000001
19	DE vs .SQP	0	0.000005	0.000003
20	DA vs .SSA	0	0.00002	0.000013
21	DE vs .JADE	0.000001	0.000079	0.000045
22	GWO vs .SCA	0.000003	0.000155	0.000087
23	PSO vs .JADE	0.000003	0.000176	0.000099
24	DE vs .MFO	0.000017	0.000918	0.000517
25	DE vs .LSHADE	0.000021	0.00117	0.00066
26	DA vs .SCA	0.00003	0.001677	0.000884
27	PSO vs .MFO	0.000034	0.001888	0.000995
28	PSO vs .LSHADE	0.000043	0.002387	0.001215
29	PSO vs .WOA	0.000055	0.003011	0.001478
30	DE vs .WOA	0.000108	0.005938	0.002699
31	PSO vs .SSA	0.001157	0.063656	0.028934
32	SCA vs .SQP	0.001398	0.076899	0.033556
33	DE vs .SSA	0.002024	0.111307	0.046546
34	DA vs .JADE	0.004863	0.267476	0.10699
35	DE vs .GWO	0.011798	0.648914	0.247767
36	PSO vs .GWO	0.01849	1.0	0.3698
37	PSO vs .SCA	0.019881	1.0	0.377745
38	GWO vs .JADE	0.021363	1.0	0.384537
39	DA vs .MFO	0.021363	1.0	0.384537
40	SSA vs .SQP	0.02294	1.0	0.384537
41	DA vs .LSHADE	0.024618	1.0	0.384537
42	DE vs .SCA	0.030302	1.0	0.424227
43	DE vs .DA	0.045106	1.0	0.586381
44	PSO vs .DA	0.065592	1.0	0.787101
45	MFO vs .GWO	0.073929	1.0	0.813223
46	GWO vs .LSHADE	0.083114	1.0	0.831139
47	WOA vs .SCA	0.088041	1.0	0.831139
48	WOA vs .SQP	0.136432	1.0	1.0
49	SSA vs .SCA	0.357258	1.0	1.0
50	SSA vs .WOA	0.432325	1.0	1.0
51	JADE vs .LSHADE	0.569623	1.0	1.0
52	MFO vs .JADE	0.606935	1.0	1.0
53	DA vs .GWO	0.606935	1.0	1.0
54	PSO vs .DE	0.870944	1.0	1.0
55	MFO vs .LSHADE	0.956813	1.0	1.0

Tabela B.3: Valores- p ajustados - CEC2017 - D=10

i	Hipótese	valor- p não ajustado	p_{Neme}	p_{Shaf}
1	DA vs .JADE	0	0	0
2	DA vs .LSHADE	0	0	0
3	WOA vs .JADE	0	0	0
4	SCA vs .JADE	0	0	0
5	WOA vs .LSHADE	0	0	0
6	DE vs .DA	0	0	0
7	SCA vs .LSHADE	0	0	0
8	MFO vs .JADE	0	0	0
9	DE vs .WOA	0	0	0
10	PSO vs .DA	0	0	0
11	DA vs .SSA	0	0.000001	0.000001
12	MFO vs .LSHADE	0	0.000002	0.000001
13	GWO vs .JADE	0	0.000004	0.000002
14	DE vs .SCA	0	0.000014	0.000009
15	PSO vs .WOA	0.000001	0.000043	0.000028
16	DA vs .GWO	0.000003	0.000156	0.000101
17	SSA vs .JADE	0.000008	0.000357	0.00023
18	SSA vs .WOA	0.000009	0.000395	0.000246
19	GWO vs .LSHADE	0.000018	0.000791	0.000422
20	PSO vs .JADE	0.00006	0.002713	0.001447
21	PSO vs .SCA	0.000079	0.003567	0.001903
22	DE vs .MFO	0.000192	0.008625	0.0046
23	WOA vs .GWO	0.000376	0.016927	0.008276
24	SSA vs .SCA	0.000481	0.021638	0.010578
25	DA vs .MFO	0.000612	0.027548	0.012856
26	SSA vs .LSHADE	0.000663	0.029831	0.013258
27	PSO vs .LSHADE	0.003187	0.143416	0.057367
28	DE vs .JADE	0.004502	0.202582	0.081033
29	GWO vs .SCA	0.009264	0.416892	0.157492
30	PSO vs .MFO	0.010504	0.472686	0.168066
31	DE vs .GWO	0.01189	0.535028	0.178343
32	WOA vs .MFO	0.019184	0.863292	0.26858
33	SSA vs .MFO	0.035431	1.0	0.4606
34	DA vs .SCA	0.041515	1.0	0.498183
35	DE vs .LSHADE	0.075382	1.0	0.829203
36	DE vs .SSA	0.103877	1.0	1.0
37	MFO vs .SCA	0.165195	1.0	1.0
38	PSO vs .GWO	0.178806	1.0	1.0
39	MFO vs .GWO	0.22462	1.0	1.0
40	PSO vs .DE	0.241613	1.0	1.0
41	DA vs .WOA	0.278264	1.0	1.0
42	JADE vs .LSHADE	0.287989	1.0	1.0
43	WOA vs .SCA	0.340023	1.0	1.0
44	SSA vs .GWO	0.373967	1.0	1.0
45	PSO vs .SSA	0.648839	1.0	1.0

Tabela B.4: Valores- p ajustados - CEC2017 - D=30

i	Hipótese	valor- p não ajustado	p_{Neme}	p_{Shaf}
1	DA vs .JADE	0	0	0
2	SCA vs .JADE	0	0	0
3	DA vs .LSHADE	0	0	0
4	WOA vs .JADE	0	0	0
5	SCA vs .LSHADE	0	0	0
6	WOA vs .LSHADE	0	0	0
7	MFO vs .JADE	0	0	0
8	DE vs .DA	0	0	0
9	PSO vs .DA	0	0	0
10	DE vs .SCA	0	0	0
11	PSO vs .SCA	0	0	0
12	MFO vs .LSHADE	0	0	0
13	DE vs .WOA	0	0	0
14	PSO vs .WOA	0	0	0
15	DA vs .SSA	0	0.000011	0.000007
16	DA vs .GWO	0	0.000022	0.000014
17	GWO vs .JADE	0.000001	0.000067	0.000043
18	SSA vs .JADE	0.000003	0.000127	0.000079
19	SSA vs .SCA	0.000005	0.000237	0.000126
20	GWO vs .SCA	0.00001	0.000437	0.000233
21	SSA vs .WOA	0.000021	0.000961	0.000513
22	WOA vs .GWO	0.000038	0.001704	0.000909
23	GWO vs .LSHADE	0.000095	0.004271	0.002088
24	DE vs .MFO	0.000135	0.006092	0.002978
25	SSA vs .LSHADE	0.000161	0.007255	0.003386
26	PSO vs .MFO	0.000161	0.007255	0.003386
27	PSO vs .JADE	0.002076	0.093399	0.03736
28	DE vs .JADE	0.002399	0.107941	0.043177
29	DA vs .MFO	0.002767	0.12453	0.047045
30	MFO vs .SCA	0.017065	0.767934	0.273043
31	PSO vs .LSHADE	0.030124	1.0	0.451862
32	SSA vs .MFO	0.030124	1.0	0.451862
33	DE vs .LSHADE	0.033579	1.0	0.451862
34	WOA vs .MFO	0.037368	1.0	0.451862
35	MFO vs .GWO	0.041515	1.0	0.456668
36	DE vs .GWO	0.075382	1.0	0.753821
37	PSO vs .GWO	0.082783	1.0	0.753821
38	DE vs .SSA	0.099347	1.0	0.79478
39	PSO vs .SSA	0.108569	1.0	0.79478
40	DA vs .WOA	0.362427	1.0	1.0
41	JADE vs .LSHADE	0.362427	1.0	1.0
42	DA vs .SCA	0.54374	1.0	1.0
43	WOA vs .SCA	0.761445	1.0	1.0
44	SSA vs .GWO	0.896482	1.0	1.0
45	PSO vs .DE	0.965407	1.0	1.0

Tabela B.5: Valores- p ajustados - CEC2017 - D=100

i	Hipótese	valor- p não ajustado	p_{Neme}	p_{Shaf}
1	SCA vs .JADE	0	0	0
2	DA vs .JADE	0	0	0
3	SCA vs .LSHADE	0	0	0
4	DA vs .LSHADE	0	0	0
5	PSO vs .SCA	0	0	0
6	PSO vs .DA	0	0	0
7	WOA vs .JADE	0	0	0
8	MFO vs .JADE	0	0	0
9	WOA vs .LSHADE	0	0	0
10	SSA vs .SCA	0	0	0
11	DA vs .SSA	0	0	0
12	PSO vs .WOA	0	0	0
13	MFO vs .LSHADE	0	0	0
14	DE vs .SCA	0	0.000002	0.000001
15	DE vs .DA	0	0.000003	0.000002
16	PSO vs .MFO	0	0.000003	0.000002
17	GWO vs .SCA	0	0.000003	0.000002
18	DA vs .GWO	0	0.000004	0.000003
19	SSA vs .WOA	0.000014	0.00065	0.000347
20	GWO vs .JADE	0.000161	0.007255	0.003869
21	SSA vs .MFO	0.000161	0.007255	0.003869
22	DE vs .JADE	0.000227	0.010236	0.005459
23	DE vs .WOA	0.000443	0.019946	0.009752
24	WOA vs .GWO	0.000612	0.027548	0.013468
25	GWO vs .LSHADE	0.003187	0.143416	0.066928
26	DE vs .MFO	0.003187	0.143416	0.066928
27	DE vs .LSHADE	0.004205	0.189225	0.07569
28	SSA vs .JADE	0.004205	0.189225	0.07569
29	MFO vs .GWO	0.004205	0.189225	0.07569
30	PSO vs .GWO	0.010504	0.472686	0.168066
31	MFO vs .SCA	0.01189	0.535028	0.178343
32	DA vs .MFO	0.013435	0.604554	0.188083
33	PSO vs .DE	0.013435	0.604554	0.188083
34	SSA vs .LSHADE	0.041515	1.0	0.498183
35	WOA vs .SCA	0.050985	1.0	0.560831
36	DA vs .WOA	0.05636	1.0	0.563602
37	PSO vs .SSA	0.099347	1.0	0.894127
38	PSO vs .JADE	0.22462	1.0	1.0
39	SSA vs .GWO	0.362427	1.0	1.0
40	DE vs .SSA	0.409932	1.0	1.0
41	JADE vs .LSHADE	0.409932	1.0	1.0
42	WOA vs .MFO	0.572892	1.0	1.0
43	PSO vs .LSHADE	0.696299	1.0	1.0
44	DE vs .GWO	0.93088	1.0	1.0
45	DA vs .SCA	0.965407	1.0	1.0

ANEXOS

I. FUNÇÕES BÁSICAS - CEC2017

As funções básicas do benchmark são definidas como:

- **Função Bent Cigar**

$$f_1(\mathbf{x}) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2 \quad (\text{I.1})$$

- **Função Zakharov**

$$f_3(\mathbf{x}) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5x_i \right)^2 + \left(\sum_{i=1}^D 0.5x_i \right)^4 \quad (\text{I.2})$$

- **Função de Rosenbrock**

$$f_4(\mathbf{x}) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2) \quad (\text{I.3})$$

- **Função de Rastrigin**

$$f_5(\mathbf{x}) = \sum_{i=1}^D (x_i^2 - 10 \cdot \cos(2\pi x_i) + 10) \quad (\text{I.4})$$

- **Função de Schaffer expandido**

$$\text{Funcao de Schaffer: } g(x, y) = 0.5 + \frac{(\sin^2(\sqrt{x^2 + y^2}) - 0.5)}{(1 + 0.001(x^2 + y^2))^2} \quad (\text{I.5})$$

$$f_6(\mathbf{x}) = g(x_1, x_2) + g(x_2, x_3) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1)$$

- **Função Lunacek bi-Rastrigin**

$$f_7(\mathbf{x}) = \min \left(\sum_{i=1}^D (\hat{x}_i - \mu_0)^2; dD + s \sum_{i=1}^D (\hat{x}_i - \mu_1)^2 + 10(D - \sum_{i=1}^D \cos(2\pi \hat{z}_i)) \right)$$

$$\mu_0 = 2.5, \mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}, s = 1 - \frac{1}{2\sqrt{D + 20} - 8.2}, d = 1 \quad (\text{I.6})$$

$$y = \frac{10(\mathbf{x} - \mathbf{o})}{100}, \hat{x}_i = 2\text{sign}(x_i^*)y_i + \mu_0, \text{ para } i = 1, 2, \dots, D$$

$$\mathbf{z} = \Lambda^{100}(\hat{\mathbf{x}} - \mu_0)$$

onde, Λ^α é uma matriz diagonal de dimensão D com o i -ésimo elemento diagonal calculado como $\lambda_{ii} = \alpha^{\frac{i-1}{2(D-1)}}$, para $i = 1, 2, \dots, D$.

• **Função de Rastrigin não contínua rotacionada**

$$f_8(\mathbf{x}) = \sum_{i=1}^D (z_i^2 - 10\cos(2\pi z_i) + 10) + f_{13}^* \quad (I.7)$$

$$\hat{x} = \mathbf{M}_1 \frac{5.12(x - o)}{100}, y_i = \begin{cases} \hat{x}_i & \text{se } |\hat{x}_i| \leq 0.5 \\ \text{round}(2\hat{x}_i)/2 & \text{se } |\hat{x}_i| > 0.5 \end{cases} \text{ para } i = 1, 2, \dots, D$$

$$z = \mathbf{M}_1 \Lambda^{10} \mathbf{M}_2 T_{asy}^{0.2}(T_{osy}(y))$$

$T_a^\beta sy$: se $x_i > 0$, $x_i = x_i^{1+\beta \frac{i-1}{D-1} \sqrt{x_i}}$, para $i = 1, 2, \dots, D$.

T_{osy} : para $x_i = \text{sign}(x_i) \exp(\tilde{x}_i + 0.049(\sin(c_1 \tilde{x}_i) + \sin(c_2 \tilde{x}_i)))$, para $i = 1, D$.

$$\text{Onde } \tilde{x}_i = \begin{cases} \log(|x_i|) & \text{se } x_i \neq 0 \\ 0 & \text{caso contrario} \end{cases}, \text{sign}(x_i) = \begin{cases} -1 & \text{se } x_i < 0 \\ 0 & \text{se } x_i = 0 \\ -1 & \text{caso contrario} \end{cases} \text{ e}$$

$$c_1 = \begin{cases} 10 & \text{se } x_i > 0 \\ 5.5 & \text{caso contrario} \end{cases} \text{ e } c_2 = \begin{cases} 7.9 & \text{se } x_i > 0 \\ 3.1 & \text{caso contrario} \end{cases}$$

• **Função de Levy**

$$f_9(\mathbf{x}) = \sin^2(\pi w_1) + \sum_{i=1}^{D-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_D - 1)^2 [1 + \sin^2(2\pi w_D)] \quad (I.8)$$

onde, $w_i = 1 + \frac{x_i - 1}{4}$, para $i = 1, 2, \dots, D$.

• **Função de Schwefel modificada**

$$f_{10}(\mathbf{x}) = 418.9828 \cdot D - \sum_{i=1}^D g(z_i), z_i = x_i + 4.209687462275036e + 02 \quad (I.9)$$

$$g(z_i) = \begin{cases} z_i \sin(|z_i|^{1/2}) & \text{se } |z_i| \leq 500 \\ (500 - \text{mod}(z_i, 500)) \sin(\sqrt{|500 - \text{mod}(z_i, 500)|}) - \frac{(z_i - 500)^2}{10000D} & \text{se } z_i > 500 \\ (\text{mod}(|z_i|, 500) - 500) \sin(\sqrt{|\text{mod}(|z_i|, 500) - 500|}) - \frac{(z_i + 500)^2}{10000D} & \text{se } z_i < -500 \end{cases}$$

• **Função Elíptica altamente condicionada**

$$f_{11}(\mathbf{x}) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} \mathbf{x}_i^2 \quad (I.10)$$

• **Função Discus**

$$f_{12}(\mathbf{x}) = 10^6 x_1^2 + \sum_{i=2}^D x_i^2 \quad (\text{I.11})$$

• **Função de Ackley**

$$f_{13}(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e \quad (\text{I.12})$$

• **Função Weierstrass**

$$f_{14}(\mathbf{x}) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k \cdot 0.5)] \quad (\text{I.13})$$

$a = 0.5, b = 3, k_{max} = 20$

• **Função de Griewank**

$$f_{15}(\mathbf{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1 \quad (\text{I.14})$$

• **Função Katsuura**

$$f_{16}(\mathbf{x}) = \frac{10}{D^2} \prod_{i=1}^D \left(1 + i \sum_{j=1}^{32} \frac{|2^j x_i - \text{round}(2^j x_i)|}{2^j} \right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2} \quad (\text{I.15})$$

• **Função HappyCat**

$$f_{17}(\mathbf{x}) = \left| \sum_{i=1}^D x_i^2 - D \right| + \left(0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i \right) / D + 0.5 \quad (\text{I.16})$$

• **Função HGBat**

$$f_{18}(\mathbf{x}) = \left| \left(\sum_{i=1}^D x_i^2 \right)^2 - \left(\sum_{i=1}^D x_i \right)^2 \right|^{1/2} + \left(0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i \right) / D + 0.5 \quad (\text{I.17})$$

• **Função de Rosenbrock com Griewank expandida**

$$f_{19}(\mathbf{x}) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + \dots + f_7(f_4(x_{D-1}, x_D)) + f_7(f_4(x_D, x_1)) \quad (\text{I.18})$$

• Função F7 de Schaffer

$$f_{20}(\mathbf{x}) = \left[\frac{1}{D-1} \sum_{i=1}^{D-1} (\sqrt{s_i} \cdot (\sin(50s_i^{0.2}) + 1)) \right]^2, \quad s_i = \sqrt{x_i^2 + x_{i+1}^2} \quad (\text{I.19})$$

II. BENCHMARK - CEC2017

- **Funções Unimodais:** F_1 a F_3

1. **Função Bent Cigar deslocada e rotacionada**

$$F_1(\mathbf{x}) = f_1(\mathbf{M}(\mathbf{x} - \mathbf{o}_1)) + F_1^* \quad (\text{II.1})$$

2. **Função excluída do benchmark**

3. **Função Zakharov deslocada e rotacionada**

$$F_3(\mathbf{x}) = f_3(\mathbf{M}(\mathbf{x} - \mathbf{o}_3)) + F_3^* \quad (\text{II.2})$$

- **Funções Multimodais:** F_4 a F_{10}

4. **Função de Rosenbrock deslocada e rotacionada**

$$F_4(\mathbf{x}) = f_4(\mathbf{M}(\frac{2.048(\mathbf{x} - \mathbf{o}_4)}{100}) + 1) + F_4^* \quad (\text{II.3})$$

5. **Função de Rastrigin deslocada e rotacionada**

$$F_5(\mathbf{x}) = f_5(\mathbf{M}(\mathbf{x} - \mathbf{o}_5)) + F_5^* \quad (\text{II.4})$$

6. **Função de Shaffer expandida, deslocada e rotacionada**

$$F_6(\mathbf{x}) = f_{20}(\mathbf{M}(\frac{0.5(\mathbf{x} - \mathbf{o}_6)}{100})) + F_6^* \quad (\text{II.5})$$

7. **Função Lunacek Bi_Rastrigin deslocada e rotacionada**

$$F_7(\mathbf{x}) = f_7(\mathbf{M}(\frac{600(\mathbf{x} - \mathbf{o}_7)}{100})) + F_7^* \quad (\text{II.6})$$

8. **Função não contínua de Rastrigin deslocada e rotacionada**

$$F_8(\mathbf{x}) = f_8(\frac{5.12(\mathbf{x} - \mathbf{o}_8)}{100}) + F_8^* \quad (\text{II.7})$$

9. **Função Levy deslocada e rotacionada**

$$F_9(\mathbf{x}) = f_9(\mathbf{M}(\frac{5.12(\mathbf{x} - \mathbf{o}_9)}{100})) + F_9^* \quad (\text{II.8})$$

10. Função de Schwefel deslocada e rotacionada

$$F_{10}(\mathbf{x}) = f_{10}(\mathbf{M}(\frac{1000(\mathbf{x} - \mathbf{o}_{10})}{100})) + F_{10}^* \quad (\text{II.9})$$

• **Funções Híbridas:** dada a equação (3.23) as funções híbridas seguem os índices de 11 a 20.

11. Função híbrida 1 (N=3)

$$p = [0.2, 0.4, 0.4]$$

g_1 : Função Zakharov f_3

g_2 : Função Rosenbrock f_4

g_3 : Função de Rastrigin f_5

12. Função híbrida 2 (N=3)

$$p = [0.3, 0.3, 0.4]$$

g_1 : Função Elíptica altamente condicionada f_{11}

g_2 : Função de Schwefel modificada f_{10}

g_3 : Função de Bent Cigar f_1

13. Função híbrida 3 (N=3)

$$p = [0.3, 0.3, 0.4]$$

g_1 : Função de Bent Cigar f_1

g_2 : Função Rosenbrock f_4

g_3 : Função Lunacek bi-Rastrigin f_7

14. Função híbrida 4 (N=4)

$$p = [0.2, 0.2, 0.2, 0.4]$$

g_1 : Função Elíptica altamente condicionada f_{11}

g_2 : Função de Ackley f_{13}

g_3 : Função F7 de Schaffer f_{20}

g_4 : Função de Rastrigin f_5

15. Função híbrida 5 (N=4)

$$p = [0.2, 0.2, 0.3, 0.3]$$

g_1 : Função de Bent Cigar f_1

g_2 : Função HGBat f_{18}

g_3 : Função de Rastrigin f_5

g_4 : Função Rosenbrock f_4

16. Função híbrida 6 (N=4)

$$p = [0.2, 0.2, 0.3, 0.3]$$

g_1 : Função de Schaffer expandido f_6

g_2 : Função HGBat f_{18}

g_3 : Função Rosenbrock f_4

g_4 : Função de Schwefel modificado f_{10}

17. Função híbrida 7 (N=5)

$$p = [0.1, 0.2, 0.2, 0.2, 0.3]$$

g_1 : Função Katsuura f_{16}

g_2 : Função de Ackley f_{13}

g_3 : Função de Rosenbrock com Griewank expandida f_{19}

g_4 : Função de Schwefel modificado f_{10}

g_5 : Função de Rastrigin f_5

18. Função híbrida 8 (N=5)

$$p = [0.2, 0.2, 0.2, 0.2, 0.2]$$

g_1 : Função Elíptica altamente condicionada f_{11}

g_2 : Função de Ackley f_{13}

g_3 : Função de Rastrigin f_5

g_4 : Função HGBat f_{18}

g_5 : Função Discus f_{12}

19. Função híbrida 9 (N=5)

$$p = [0.2, 0.2, 0.2, 0.2, 0.2]$$

g_1 : Função Bent Cigar f_1

g_2 : Função de Rastrigin f_5

g_3 : Função de Rosenbrock com Griewank expandida f_{19}

g_4 : Função Weierstrass f_{14}

g_5 : Função de Schaffer expandido f_6

20. Função híbrida 10 (N=6)

$$p = [0.1, 0.1, 0.2, 0.2, 0.2, 0.2]$$

g_1 : Função Happycat f_{17}

g_2 : Função Katsuura f_{16}

g_3 : Função de Ackley f_{13}

g_4 : Função de Rastrigin f_5

g_5 : Função de Schwefel modificado f_{10}

g_6 : Função F7 de Schaffer f_{20}

• **Funções Compostas:**

21. **Função composta 1 (N=3)**

$$\sigma = [10, 20, 30]$$

$$\lambda = [1, 1e - 6, 1]$$

$$\text{bias} = [0, 100, 200]$$

g_1 : Função de Rosenbrock f_4

g_2 : Função Elíptica altamente condicionada f_{11}

g_3 : Função de Rastrigin f_5

22. **Função composta 2 (N=3)**

$$\sigma = [10, 20, 30]$$

$$\lambda = [1, 10, 1]$$

$$\text{bias} = [0, 100, 200]$$

g_1 : Função de Rastrigin f_5

g_2 : Função de Griewank f_{15}

g_3 : Função de Schwefel modificado f_{10}

23. **Função composta 3 (N=4)**

$$\sigma = [10, 20, 30, 40]$$

$$\lambda = [1, 10, 1, 1]$$

$$\text{bias} = [0, 100, 200, 300]$$

g_1 : Função de Rosenbrock f_4

g_2 : Função de Ackley f_{13}

g_3 : Função de Schwefel modificado f_{10}

g_4 : Função de Rastrigin f_5

24. **Função composta 4 (N=4)**

$$\sigma = [10, 20, 30, 40]$$

$$\lambda = [10, 1e - 6, 10, 1]$$

$$\text{bias} = [0, 100, 200, 300]$$

g_1 : Função de Ackley f_{13}

g_2 : Função Elíptica altamente condicionada f_{11}

g_3 : Função de Griewank f_{15}

g_4 : Função de Rastrigin f_5

25. **Função composta 5 (N=5)**

$\sigma = [10, 20, 30, 40, 50]$
 $\lambda = [10, 1, 10, 1e - 6, 1]$
 $\text{bias} = [0, 100, 200, 300, 400]$
 g_1 : Função de Rastrigin f_5
 g_2 : Função Happycat f_{17}
 g_3 : Função de Ackley f_{13}
 g_4 : Função Discus f_{12}
 g_5 : Função de Rosenbrock f_4

26. Função composta 6 (N=5)

$\sigma = [10, 20, 30, 40, 50]$
 $\lambda = [1e - 26, 10, 1e - 6, 10, 5e - 4]$
 $\text{bias} = [0, 100, 200, 300, 400]$
 g_1 : Função de Schaffer expandida f_6
 g_2 : Função de Schwefel modificada f_{10}
 g_3 : Função de Griewank f_{15}
 g_4 : Função de Rosenbrock f_4
 g_5 : Função de Rastrigin f_5

27. Função composta 7 (N=6)

$\sigma = [10, 20, 30, 40, 50, 60]$
 $\lambda = [10, 10, 2.5, 1e - 26, 1e - 6, 5e - 4]$
 $\text{bias} = [0, 100, 200, 300, 400, 500]$
 g_1 : Função HGBat f_{18}
 g_2 : Função de Rastrigin f_5
 g_3 : Função de Schwefel modificada f_{10}
 g_4 : Função Bent Cigar f_1
 g_5 : Função Elíptica altamente condicionada f_{11}
 g_6 : Função de Schaffer expandida f_6

28. Função composta 8 (N=6)

$\sigma = [10, 20, 30, 40, 50, 60]$
 $\lambda = [10, 10, 1e - 6, 1, 1, 5e - 4]$
 $\text{bias} = [0, 100, 200, 300, 400, 500]$
 g_1 : Função de Ackley f_{13}
 g_2 : Função de Griewank f_{15}
 g_3 : Função Discus f_{12}
 g_4 : Função de Rosenbrock f_4
 g_5 : Função Happycat f_{17}

g_6 : Função de Schaffer expandida f_6

29. Função composta 9 (N=3)

$\sigma = [10, 30, 50]$

$\lambda = [1, 1, 1]$

bias = [0,100,200]

g_1 : Função híbrida 5 F_5

g_2 : Função híbrida 6 F_6

g_3 : Função híbrida 7 F_7

30. Função composta 10 (N=3)

$\sigma = [10, 30, 50]$

$\lambda = [1, 1, 1]$

bias = [0,100,200]

g_1 : Função híbrida 5 F_5

g_2 : Função híbrida 8 F_8

g_3 : Função híbrida 9 F_9