

UNIVERSIDADE FEDERAL DO PARANÁ

FLAVIA BERNARDO PINTO

METAHEURÍSTICAS DE OTIMIZAÇÃO APLICADAS À SINTONIA DOS
GANHOS DE CONTROLADOR PID MULTIVARIÁVEL

CURITBA

2016

FLAVIA BERNARDO PINTO

METAHEURÍSTICAS DE OTIMIZAÇÃO APLICADAS À SINTONIA DOS GANHOS
DE CONTROLADOR PI MULTIVARIÁVEL

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Engenharia Elétrica, no Programa de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná.

Orientador: Prof. Leandro dos Santos Coelho,
Dr.

CURITIBA

2016

FOLHA DE APROVAÇÃO

FLAVIA BERNARDO PINTO

METAHEURÍSTICAS DE OTIMIZAÇÃO APLICADAS À SINTONIA DOS GANHOS DE CONTROLADOR PI MULTIVARIÁVEL

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, pela seguinte banca examinadora:



Prof. Dr. Gideon Villar Leandro

Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), UFPR



Prof. Dr. Roberto Zanetti Freire

Programa de Pós-Graduação em Eng. de Produção e Sistemas (PPGEPS), PUCPR



Prof. Dr. Leandro dos Santos Coelho

Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), UFPR

Curitiba, 04 de maio de 2016.

AGRADECIMENTOS

À meu orientador, Prof. Dr. Leandro dos Santos Coelho, pelo acompanhamento, orientação e dedicação.

À minha família pelo apoio e compreensão.

RESUMO

Esta dissertação tem por objetivo avaliar abordagens de sintonia de controladores PI (Proporcional e Integral) multivariável e acoplado, utilizando metaheurísticas de otimização aplicada a soma ponderada dos sinais de erro do sistema. Os controladores PI e PID (Proporcional, Integral e Derivativo) são os controladores mais utilizados na indústria, pois possui um algoritmo simples e eficiente. Nesta dissertação, o algoritmo evolutivo denominado evolução diferencial (DE), é comparado a outros algoritmos derivados do DE clássico e também a outros algoritmos evolutivos baseados em população. Estes algoritmos são aplicados na otimização de controle PI em dois estudos de caso: um processo de uma caldeira de turbina (*Boiler-Turbine*) e um processo de controle de nível (*Quadruple Tank*). O processo de otimização lida com a soma ponderada dos sinais de erro dos sistemas tratando-os como um problema de otimização mono-objetivo. Nos dois estudos de caso o algoritmo que obteve o melhor desempenho entre todos os algoritmos foi o EPSDE (*Ensemble of Mutation and Crossover Strategies and Parameters in DE*), e o que apresentou o desempenho menos promissor entre todos os algoritmos testados foi o CMAES (do inglês, *Covariance Matrix Adaptation Evolution Strategy*). Entre os algoritmos baseados em população o que apresentou o pior desempenho nos dois estudos de caso foi o MVO (do inglês, *Multi-Verse Optimization*) e o que apresentou o melhor desempenho foi PSO (do inglês, *Particle Swarm Optimization*). Para o primeiro estudo de caso, o DE clássico teve um bom desempenho, o que não ocorreu no segundo estudo de caso. Os algoritmos variantes de DE apresentaram um bom desempenho para os dois estudos de caso quando comparados a outros algoritmos baseados em população aplicados nesta dissertação, concluindo assim, a eficácia dos algoritmos DE para os casos testados.

Palavras-chave: Controle PI Multivariável, Metaheurísticas de Otimização, Algoritmo de Evolução Diferencial.

ABSTRACT

This thesis focuses on validate the approaches used for PI control (proportional and integral) multivariable and coupled using metaheuristics optimization applied the weighted sum of the system error signals. The PI and PID (proportional, integral and derivative) controllers are the controllers most commonly used in the industry because it has a simple and efficient algorithm. In this thesis the evolutionary algorithm named differential evolution (DE) is compared to other derived algorithms and also other evolutionary algorithms based on population. These algorithms are applied to the optimization of PI control in two case studies: a process of a boiler turbine (Boiler-Turbine) and level control process (Quadruple Tank). The optimization process deals with the weighted sum of the systems errors signals by treating them as a single-objective optimization problem. In the two case studies the algorithm which obtained the best performance among all algorithms was the EPSDE (Ensemble of Mutation and Crossover Strategies and Parameters in DE) and presented the performance less promising among all algorithms tested was the CMAES (Covariance Matrix Adaptation Evolution Strategy). Among the algorithms based on population presented the worst performance in two case studies was the MVO (Multi-Verse Optimization) and presented the best performance was PSO (Particle Swarm Optimization). For the first case study, the classic DE had a good performance, which did not occur in the second case study. The algorithms DE variants performed well for the two case studies compared to other based population algorithms applied in this thesis, concluding thus the effectiveness of DE algorithms for the cases tested.

Key-words: PI Control Multivariable, Optimization Metaheuristics, Differential Evolutionary Algorithm.

LISTA DE ILUSTRAÇÕES

FIGURA 1.1 - CLASSIFICAÇÃO DOS MÉTODOS ESTOCÁSTICOS	20
FIGURA 1.2 - CONTROLE PI DESACOPLADO	25
FIGURA 1.3 - CONTROLE PI ACOPLADO.....	26
FIGURA 2.1 - CICLO DE OTIMIZAÇÃO DE UM ALGORITMO EVOLUTIVO ..	28
FIGURA 2.2 - ALGORITMO DE EVOLUÇÃO DIFERENCIAL.....	32
FIGURA 2.3 - ESQUEMA NO ESPAÇO 2-D DO VETOR DE PERTURBAÇÃO	33
FIGURA 2.4 - PSEUDO-CÓDIGO DO ALGORITMO PSO.....	40
FIGURA 2.5 - PSEUDO-CÓDIGO DO ALGORITMO MVO	41
FIGURA 2.6 - HIERARQUIA DE LOBOS CINZENTOS	42
FIGURA 2.7 - PSEUDO-CÓDIGO DO ALGORITMO GWO	43
FIGURA 2.8 - PSEUDO-CÓDIGO DO ALGORITMO ALO	44
FIGURA 2.9 - PSEUDO-CÓDIGO DO ALGORITMO MODA	46
FIGURA 2.10 - SOLUÇÃO DE DOMINÂNCIA	48
FIGURA 2.11 - METAS DA OTIMIZAÇÃO MULTIOBJETIVO.....	49
FIGURA 3.1 - ESQUEMA DO BOILER TURBINE.....	55
FIGURA 3.2 - COMPARATIVO DAS SAÍDAS DO <i>BOILER-TURBINE</i> DOS ESTUDOS I, II, III, E IV PARA DE CLÁSSICO.....	58
FIGURA 3.3 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS 4 TESTES - <i>SISTEMA BOILER TURBINE</i>	59
FIGURA 3.4 - COMPARATIVO ENTRE OS ALGORITMOS DE CLÁSSICO E SUAS RAMIFICAÇÕES – <i>SISTEMA BOILER TURBINE</i>	61
FIGURA 3.5 - COMPARATIVO ENTRE OS ALGORITMOS DE CLÁSSICO E ALGORITMOS BASEADOS EM POPULAÇÃO - <i>SISTEMA BOILER TURBINE</i>	61
FIGURA 3.6 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS ALGORITMOS DERIVADOS DO DE - <i>SISTEMA BOILER TURBINE</i>	62
FIGURA 3.7 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS ALGORITMOS DERIVADOS DO DE SEM O CMAES - <i>SISTEMA BOILER TURBINE</i>	63

FIGURA 3.8 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DO ALGORITMO DE COMPARADO AOS ALGORITMOS BASEADOS EM POPULAÇÃO – SISTEMA <i>BOILER TURBINE</i>	64
FIGURA 3.9 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DO ALGORITMO DE COMPARADO AOS ALGORITMOS BASEADOS EM POPULAÇÃO SEM MVO – SISTEMA <i>BOILER TURBINE</i>	65
FIGURA 3.10 - ESQUEMA DO <i>QUADRUPLE TANK</i>	66
FIGURA 3.11 - COMPARATIVO DA SAÍDA DO <i>QUADRUPLE TANK</i> DOS TESTES I, II, III E IV PARA DE CLÁSSICO	70
FIGURA 3.12 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS 4 ESTUDOS - SISTEMA <i>BOILER TURBINE</i>	71
FIGURA 3.13 - COMPARATIVO ENTRE OS ALGORITMOS DE CLÁSSICO E SUAS RAMIFICAÇÕES – SISTEMA <i>QUADRUPLE TANK</i>	73
FIGURA 3.14 - COMPARATIVO ENTRE OS ALGORITMOS DE CLÁSSICO E ALGORITMOS BASEADOS EM POPULAÇÃO - SISTEMA <i>QUADRUPLE TANK</i>	73
FIGURA 3.15 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS ALGORITMOS DERIVADOS DO DE - SISTEMA <i>QUADRUPLE TANK</i>	75
FIGURA 3.16 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS ALGORITMOS DERIVADOS DO DE SEM O CMAES - SISTEMA <i>QUADRUPLE TANK</i>	75
FIGURA 3.17 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DO ALGORITMO DE COMPARADO AOS ALGORITMOS BASEADOS EM POPULAÇÃO – SISTEMA <i>QUADRUPLE TANK</i>	77
FIGURA 3.18 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DO ALGORITMO DE COMPARADO AOS ALGORITMOS BASEADOS EM POPULAÇÃO SEM MVO – SISTEMA <i>QUADRUPLE TANK</i>	77
FIGURA A2.0.1 - COMPARAÇÃO ENTRE OS MELHORES RESULTADOS PARA <i>BOILER TURBINE</i>	93
FIGURA A2.0.2 - COMPARAÇÃO ENTRE OS MELHORES RESULTADOS PARA <i>QUADRUPLE TANK</i>	94

LISTA DE TABELAS

TABELA 2.1 - TABELA DE ESTRATÉGIAS DA DE.....	31
TABELA 3.1 - VALORES DO FATOR DE CRUZAMENTO E MUTAÇÃO UTILIZADOS – SISTEMA <i>BOILER</i>	54
TABELA 3.2 - VALORES INICIAIS PARA AS VARIÁVEIS DE ENTRADA, CONTROLE E SAÍDA	57
TABELA 3.3 - VALORES DE MÍNIMA E MÁXIMA FUNÇÃO OBJETIVO E DESVIO PADRÃO PARA CADA TESTE – SISTEMA <i>BOILER TURBINE</i>	58
TABELA 3.4 - COMPARATIVO POR WILCOXON-MANN-WHITNEY U-TEST ENTRE OS TESTES – SISTEMA <i>BOILER TURBINE</i>	59
TABELA 3.5 - MÍNIMO, MÁXIMO, MÉDIA E DESVIO PADRÃO ENTRE ALGORITMOS - SISTEMA <i>BOILER TURBINE</i>	60
TABELA 3.6 - TESTE T – DE CLÁSSICO VERSUS ALGORITMOS DE RAMIFICADOS – SISTEMA <i>BOILER TURBINE</i>	62
TABELA 3.7-TESTE	62
TABELA 3.8 - TESTE T – DE CLÁSSICO VERSUS OUTROS ALGORITMOS BASEADOS EM POPULAÇÃO - SISTEMA <i>BOILER TURBINE</i>	64
TABELA 3.9 - VALORES DAS CARACTERÍSTICAS DOS TANQUES E AMBIENTE	67
TABELA 3.10 - VALORES DE CONDIÇÕES INICIAIS DO SISTEMA DE <i>QUADRUPLE TANK</i>	68
TABELA 3.11 - VALORES DE MÍNIMA E MÁXIMA FUNÇÃO OBJETIVO E DESVIO PADRÃO PARA CADA TESTE - SISTEMA <i>QUADRUPLE TANK</i>	69
TABELA 3.12 - COMPARATIVO POR WILCOXON-MANN-WHITNEY U-TEST ENTRE OS TESTES – SISTEMA <i>QUADRUPLE TANK</i>	71
TABELA 3.13 - MÍNIMO, MÁXIMO, MÉDIA E DESVIO PADRÃO ENTRE ALGORITMOS - SISTEMA <i>QUADRUPLE TANK</i>	72
TABELA 3.14 - TESTE DE WILCOXON-MANN-WHITNEY U-TEST – DE CLÁSSICO VERSUS ALGORITMOS DE RAMIFICADOS – SISTEMA <i>QUADRUPLE TANK</i>	74

TABELA 3.15 - TESTE DE WILCOXON-MANN-WHITNEY U-TEST – DE CLÁSSICO VERSUS OUTROS ALGORITMOS BASEADOS EM POPULAÇÃO - SISTEMA <i>QUADRUPLE TANK</i>	76
TABELA A.1 - TESTE T – ALGORITMOS DERIVADOS DO DE – <i>BOILER TURBINE</i>	89
TABELA A.2 - TESTE T – ALGORITMOS BASEADOS EM POPULAÇÃO – <i>BOILER TURBINE</i>	90
TABELA A.3 - TESTE WILCOXON – ALGORITMOS DERIVADOS DO DE – <i>QUADRUPLE TANK</i>	91
TABELA A.4 - TESTE WILCOXON – ALGORITMOS BASEADOS EM POPULAÇÃO – <i>QUADRUPLE TANK</i>	92

LISTA DE SIGLAS

ADE	- <i>Adaptive Differential Evolution</i>
ALO	- <i>Ant Lion Optimization</i>
AWFPID	- <i>Adaptive Wavelet Fuzzy PID Control</i>
BDA	- <i>Binary Dragonfly Algorithm</i>
CMA	- <i>Covariance Matrix Adaptation</i>
CMAES	- <i>Covariance Matrix Adaptation Evolution Strategy</i>
CODE	- <i>Composite Differential Evolution</i>
DA	- <i>Dragonfly Algorithm</i>
DE	- <i>Differential Evolution</i>
DNA	- <i>Deoxyribonucleic Acid</i>
DESAP	- <i>Differential Evolution with Self-Adapting Populations</i>
EA	- <i>Evolutionary Algorithm</i>
EPSDE	- <i>Ensemble of Mutation and Crossover Strategies and Parameters in Differential Evolutionary</i>
GA	- <i>Genetic Algorithm</i>
GWO	- <i>Grey Wolf Optimization</i>
HJPSO	- <i>Hybrid Jump Particle Swarm Optimization</i>
HMPC	- <i>Hierarchical Model Predictive Control</i>
IAE	- <i>Integral of the Absolute Magnitude of the Error</i>
ISE	- <i>Integral of Square of the Error</i>
ITAE	- <i>Integral Time-Weighted Absolute Error</i>
ITSE	- <i>Integral of Time Multiply Square Error</i>
JADE	- <i>Adaptive Differential evolution with optional External Archive</i>
jDE	- <i>j-Differential Evolution</i>
LSGO	- <i>Large Scale Global Optimization</i>
MIMO	- <i>Multiple Input, Multiple Output</i>
MLS	- <i>Multi-Level Switching</i>
MODA	- <i>Multi-Objective Dragonfly Algorithm</i>
MOP	- <i>Multi-Objective Optimization</i>
mpMPC	- <i>Multi-Parametric Model Predictive Controller</i>

- MVO - *Multi-Verse Optimization*
- PD - *Proportional-Derivative*
- PID - *Proportional-Integral-Derivative*
- PSO - *Particle Swarm Optimization*
- RASMC - *Robust Adaptive Sliding Mode Controller*
- SaDE - *Self-adaptive Differential Evolution*
- SISO - *Single Input, Single Output*
- SLV - *Satellite Launch Vehicle*
- SOP - *Single-Objective Optimization*

LISTA DE SÍMBOLOS

A_i	- Área de Seção dos Tanques (cm ²)
B_g	- Matriz Ortogonal dos Autovalores de C_g
C_g	- Matriz de Covariância
Cr	- Constante de <i>Crossover</i>
Cr_i	- Constante de <i>Crossover</i> no JADE
Cr_m	- Mediana do Conjunto Cr Depois do Período de Aprendizado
D	- Número de variáveis
Dz_g	- Matriz Diagonal das Raízes Quadradas dos Autovalores de C_g
$E(s)$	- Sinal de Erro
F	- Constante de Mutação
F_i	- Constante de Mutação no JADE
$F_{L,g}$	- Valor Mínimo do Parâmetro da Constante de Mutação
$F_{U,g}$	- Valor Máximo do Parâmetro da Constante de Mutação
H_0	- Hipótese Nula
H_1	- Hipótese Alternativa
K_d	- Ganho Derivativo
K_i	- Ganho Integrativo
K_p	- Ganho Proporcional
$N(0, I)$	- Distribuição Normal com Média Zero e Matriz de Covariância Unitária
NP	- Tamanho da População
P_0	- Potência de Saída (MW)
$P_{\sigma,g}$	- Caminho da Evolução Conjugada
R_A^t	- Caminho Aleatório em Torno da Formiga-Leão
R_E^t	- Caminho Aleatório em Torno da Elite
R^D	- Espaço de Busca
S_{Cr}	- Taxas de Todos os Cruzamentos
S_F	- Taxa de Todas as Mutações
T	- Tempo de Amostragem (s)

T_d	-	Constante de Tempo Derivativo (s)
T_i	-	Constante de Tempo Integral (s)
$U(s)$	-	Sinal de Controle
Y	-	Espaço Objetivo
$Y(s)$	-	Saída do Sistema
X_i^t	-	Caminho Aleatório das Formigas
X	-	Espaço de Parâmetros
X_w	-	Nível da Água (m)
t	-	Período de Tempo
a_i	-	Área de Seção dos Furos dos Tanques (cm ²)
c	-	Constante Positiva no Intervalo de (0,1)
c_1	-	Atração Responsável pela Parcela Cognitiva
c_2	-	Atração Responsável pela Parcela Social
c^t	-	Mínimo de Todas as Variáveis da t Iteração
$e(k)$	-	Erro no Instante Atual
$e(k-1)$	-	Erro no Instante Anterior
$e(k-2)$	-	Erro Dois Passos Passado
$e(t)$	-	Erro da Saída
$f(k)$	-	Função em Dado Instante
$f(x)$	-	Função Objetivo
g	-	Aceleração da Gravidade (cm/s ²)
gen_{max}	-	Número Máximo de Gerações
$gbest_i$	-	Número de Partículas Vizinhas da Posição que Resulta o Melhor <i>fitness</i>
$g_i(x)$	-	Inequações do Problema
i	-	Nível dos Tanques (cm)
$i(x)$	-	Equações de Restrição do Problema
k	-	Número de Funções Objetivo
$k_i u_i$	-	Vazão (m ³ /s)
m	-	Quantidade de Inequações
$mean_A$	-	Média Aritmética Atual
$mean_L$	-	Média Lehmer
m_g	-	Média para o Algoritmo CMAES

$\min_x f(x)$	-	Equação de Minimização
p	-	Pressão do Cilindro (kg/cm ²)
$pbest_i$	-	Posição Prévia com Melhor <i>fitness</i>
p_f	-	Densidade do Fluido (kg/m ³)
q_e	-	Evaporação (kg/s)
q_0	-	Primeira parcela discretizada do controlador
q_1	-	Segunda parcela discretizada do controlador
q_2	-	Parcela Derivativa discretizada do controlador
$randc_i$	-	Distribuição de Cauchy
$randn_i$	-	Distribuição Normal
$rand_k$	-	Distribuição aleatória no intervalo [0,1]
r_i	-	Peso para a Recombinação
t	-	Tempo (s)
ud e UD	-	Distribuição Uniforme
u_i	-	Tensão nas Válvulas de Controle de Vazão (V)
$u_{i,g}$	-	Nova População
$u(k-1)$	-	Valor da ação integral no instante anterior
u_1	-	Válvula de posição do fluxo de combustível
u_2	-	Válvula de posição do controle de vapor
u_3	-	Válvula de posição do fluxo da água de alimentação
$v_{i,g}$	-	Vetor de Perturbação
v_{id}	-	Velocidade em PSO
v_1 e v_2	-	Tensão nas Válvulas (V)
x	-	Vetor de decisão
x_{best}	-	Melhor Indivíduo
$x_{best,g}$	-	Melhor Indivíduo da geração
x_{id}	-	Posição no PSO
$x_{i,L}$	-	Limites Inferiores das Variáveis de Projeto
$x_{i,U}$	-	Limites Superiores das Variáveis de Projeto
$x_{i,j}$	-	População Inicial
$x_{i,g+1}$	-	Indivíduo da nova Geração
$x_{r_j^i,g}$	-	Vetores selecionados aleatoriamente entre [1, NP]

$x(k)$	-	Valor Pretendido
$x(k-1)$	-	Valor Atrasado
w	-	Ponderação de Inércia
$w_{i,g}$	-	Melhor Indivíduo da População
y_1 e y_2	-	Nível de água dos tanques inferiores (cm)
α	-	Nível de significância
α_{cs}	-	Qualidade do vapor
μ_{Cr}	-	Média da Constante de Cruzamento
μ_F	-	Média da Mutação
τ_1	-	Responsável pela constante de Mutação
τ_2	-	Responsável pela constante de Cruzamento
σ_g	-	Passo de C_g

SUMÁRIO

1 INTRODUÇÃO	19
1.1 OBJETIVOS	22
1.1.1 Objetivos Específicos	23
1.2 CONTROLE PID MULTIVARIÁVEL	23
1.2.1 Modelo do Controlador Digital	24
1.2.2 Controle PID Multivariável Acoplado	25
2 ALGORITMOS EVOLUTIVOS	27
2.1 CARACTERÍSTICAS DOS ALGORITMOS EVOLUTIVOS	27
2.2 RESTRIÇÕES	28
2.3 ALGORITMO DE EVOLUÇÃO DIFERENCIAL.....	29
2.3.1 Estratégias do Algoritmo de Evolução Diferencial.....	30
2.3.2 Algoritmo de Evolução Diferencial Clássico (DE).....	31
2.3.3 Algoritmo CMAES (<i>Covariance Matrix Adaptation</i> Evolution Strategy).....	34
2.3.4 Algoritmo CODE (<i>Composite Differential Evolution</i>).....	35
2.3.5 Algoritmo EPSDE (<i>Ensemble of Mutation and Crossover Strategies and Parameters in Differential Evolution</i>)	36
2.3.6 Algoritmo JADE (<i>Adaptive Differential Evolution with Optional External Archive</i>)	36
2.3.7 Algoritmo jDE (<i>j-Differential Evolution</i>)	37
2.3.8 Algoritmo SaDE (<i>Self-adaptive Differential Evolution</i>).....	38
2.4 OUTROS ALGORITMOS BASEADOS EM POPULAÇÃO	39
2.4.1 Algoritmo PSO (<i>Particle Swarm Optimization</i>)	39
2.4.2 Algoritmo MVO (<i>Multi Verse Optimizer</i>)	40
2.4.3 Algoritmo GWO (<i>Grey Wolf Optimizer</i>).....	42
2.4.4 Algoritmo ALO (<i>Ant Lion Optimizer</i>)	43

2.4.5	Algoritmo DA (Dragonfly Algorithm)	45
2.5	OTIMIZAÇÃO MULTIOBJETIVO.....	47
2.5.1	Definição de Pareto-ótimo.....	47
2.5.2	Classificação de um Problema de Otimização Multiobjetivo	49
3	DESCRIÇÃO DOS ESTUDOS DE CASO	51
3.1	TESTE DE WILCOXON-MANN-WHITNEY U-TEST	51
3.2	ÍNDICES DE DESEMPENHO.....	52
3.3	FUNÇÃO OBJETIVO.....	53
3.4	PARÂMETROS DE SIMULAÇÃO	54
3.5	ESTUDO DE CASO 1 – <i>BOILER AND TURBINE</i>	55
3.5.1	Resultados DE Clássico.....	57
3.5.2	Resultados de Algoritmos Variantes do DE e outros algoritmos	60
3.6	ESTUDO DE CASO 2 – <i>QUADRUPLE TANK</i>	65
3.6.1	Resultados usando algoritmo DE clássico	69
3.6.2	Resultados de Algoritmos Variantes do DE e outros algoritmos	72
4	CONCLUSÃO	78
	REFERÊNCIAS.....	79
	APÊNDICES	88

1 INTRODUÇÃO

Muitos problemas de Engenharia, principalmente os presentes no meio industrial, são multiobjetivo e/ou não-lineares, sendo, muitas vezes, classificados como sistemas complexos difíceis de modelar, controlar e atender as restrições impostas. Assim, de acordo com REYNOSO-MEZA *et al.* (2014), as técnicas de otimização multiobjetivo acabaram ganhando destaque nos últimos anos, pois buscam um conjunto de soluções potencialmente promissoras ou mesmo “ótimas”, apresentando facilidade em incorporar restrições e outras características na formulação do problema. Alguns exemplos de utilização de técnicas de otimização multiobjetivo podem ser vistas em casos de: identificação automática do comportamento do vento em turbinas eólicas (CAVA *et al.*, 2016), bioprocessos (XU *et al.*, 2016), controle de processos de óleo-gás (LIU *et al.*, 2015), aprendizado em neurocomputação (LI; KWONG, 2014), controle preditivo baseado em modelo (ASCIONE *et al.*, 2016) e em abordagens multiobjetivo usando algoritmos de otimização para sintonia de controladores do tipo PID (Proporcional, Integral e Derivativo) (SAHIB *et al.*, 2016).

Muitos são os métodos desenvolvidos para atender um projeto de otimização, entre eles estão as abordagens de busca divididas em: Métodos Determinísticos, Enumerativos e Estocásticos.

Neste contexto, os métodos estocásticos, também conhecidos como Metaheurísticas de Otimização, de acordo com Boussaïd *et al.* (2013) são largamente reconhecidos como abordagens eficientes para solução de muitos problemas de otimização que não possuem solução exata. Os métodos estocásticos incluem entre outros, abordagens bioinspiradas (baseadas nos princípios da Biologia e Física) de componentes estocásticos (variáveis aleatórias), e geralmente, possuem poucos parâmetros que precisam ser ajustados.

As metaheurísticas são classificadas em: baseadas em população e baseadas em uma única solução conforme exibido na FIGURA 1.1, a qual mostra a classificação de alguns algoritmos. O algoritmo de evolução

diferencial (do inglês, *Differential Evolution*, DE), um dos algoritmos abordados nesta dissertação, é algoritmo evolutivo classificado como algoritmo evolutivo.

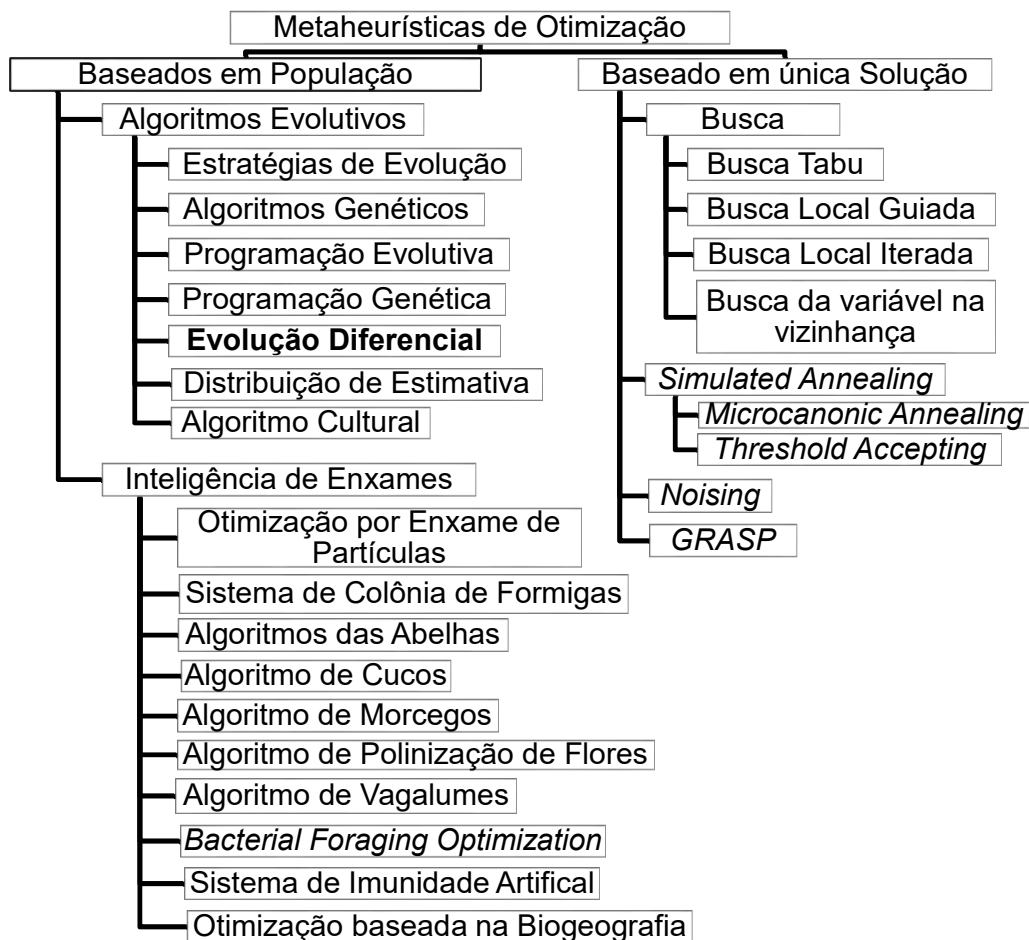


FIGURA 1.1 - CLASSIFICAÇÃO DOS MÉTODOS ESTOCÁSTICOS
 FONTE: DO AUTOR (2015)

A busca por soluções nos algoritmos evolutivos (do inglês, *Evolutionary Algorithms*, EAs) partem de regras de probabilidade de forma estocástica, não dependendo de informações relacionadas a cálculos de derivadas das funções objetivo (DELBEM; GABRIEL, 2016), aumentando assim, a chance de encontrar uma solução promissora.

No contexto de AEs, os algoritmos de DE utilizam a diferença de vetores gerados aleatoriamente que sofrerão mutação, têm fácil implementação, são eficazes em populações pequenas, possuem poucos parâmetros de controle, são robustos, tem boas propriedades de convergência, possuem

capacidade de lidar com funções não-lineares, não diferenciáveis e multimodais (BOCCATO *et al.*, 2009). Nos últimos anos, abordagens de DE têm conseguido destaque em trabalhos de otimização, além de apresentarem bons, e em alguns casos até melhores, resultados do que muitos outros algoritmos bioinspirados (DAS *et al.*, 2009, citado por SEGUNDO, 2012).

Muitas aplicações do DE têm sido pesquisadas e/ou aprimoradas nos últimos anos como é o caso de problemas de: otimização da trajetória de um satélite (Satellite Launch Vehicle, SLV) (GHOSH; CHATTOPADHYAY, 2015), controlabilidade multiobjetiva robusta de uma rede neural complexa (TANG *et al.*, 2015), otimização multimodal (ZHANG *et al.*, 2015), otimização de parâmetros paralelos contínuos (IACCA *et al.*, 2015). De acordo com Venske (2014) na DE clássica os parâmetros de controle do algoritmo e as estratégias são pré-definidos e não são alterados durante o processo evolutivo. Por outro lado, na Evolução Diferencial Adaptativa (do inglês, *Adaptive Differential Evolution*, ADE) os parâmetros e estratégias são alterados dinamicamente com o processo evolutivo.

Muitas são as aplicações do algoritmo DE, como às presentes em: refrigeração por compressão (PAIVA, 2011), problema de programação não linear (MELO *et al.*, 2014), solução de sistemas lineares (TSAI *et al.*, 2011); otimização de processos químicos (BABU; ANGIRA, 2006, citado por OLIVEIRA, 2006), otimização combinatória (KRAUSE *et al.*, 2014); planejamento hidrelétrico (BERBERT *et al.*, 2014), seleção de parâmetros de processos de manufatura (YILDIZ, 2013), mapeamento do tempo de consumo de multitarefas em máquinas virtuais (FALCO *et al.*, 2015) e controle do fluxo de potência (ACHARJEE, 2016).

O DE por conseguir tratar as várias entradas e saídas do sistema, está entre as alternativas utilizadas na otimização multiobjetivo. Processos químicos (BHASKAR, 2014), processos de controle de poluição (MARINGANTI, 2011) e soldagem (GOMES, 2011), são alguns exemplos de sistemas MIMO (do inglês, *Multiple Input, Multiple Output*).

Um sistema é considerado MIMO quando há mais de uma variável de entrada e mais de uma variável de saída que deverão ser controladas simultaneamente. Muitas vezes, devido à presença de acoplamentos suas

variáveis se interagem e afetam outras variáveis do processo, tornando-se geralmente, um sistema complexo de ser modelado e controlado.

Muitos processos industriais com estrutura MIMO utilizam os controladores PID pelo fato do fácil manuseio de seus parâmetros, por apresentar um algoritmo robusto e versátil e de baixo custo, como por exemplo: VLACHOS *et al.* (2002), utilizam um algoritmo genético para controlar dois controladores PID atuantes em coluna de destilação tipo Shell, LI *et al.* (2015) que desenvolve para o controlador PD (Proporcional e Derivativo) uma regra baseada no controle nebuloso (*fuzzy*) aplicado ao controle de velocidade do compressor do condicionador de ar para controle de temperatura e umidade, YANG *et al.* (2016) propõem uma rede neural baseada na estratégia do controle preditivo aplicado ao controle de temperatura e umidade em condicionadores de ar, VELASCO *et al.* (2016) propuseram um esquema global de controlador PID, denominado de AWFPID (do inglês, *Adaptive Wavelet Fuzzy PID control*) aplicado a sistema não-linear MIMO de uma interface robótica. Porém, geralmente, quanto mais complexo for o sistema, mais demorado é o ajuste dos parâmetros.

Assim, o DE foi escolhido para ajustar os parâmetros de controladores do tipo PI (Proporcional e Integral), pois tem a capacidade de otimizar sistemas não-lineares, são recomendados para sistemas complexos (sistemas que possuem grande variedade de parâmetros e características), está apresentando bom desempenho nos últimos artigos e funcionam tanto em sistemas contínuos como sistemas discretos.

1.1 OBJETIVOS

O objetivo geral é otimizar os ganhos de controladores do tipo PI, quando aplicados em processos multivariáveis, utilizando metaheurísticas de otimização baseadas em população.

1.1.1 Objetivos Específicos

Os objetivos específicos são:

Analisar as estratégias do algoritmo de evolução diferencial e algumas de suas variantes;

Analisar os controladores PI multivariáveis quando da aplicação de otimização usando evolução diferencial;

Simular e testar o algoritmo clássico de DE aplicado a dois estudos de caso;

Comparar o desempenho de diferentes métodos DE nos estudos de caso adotados.

1.2 CONTROLE PID MULTIVARIÁVEL

De acordo com Iruthayarajan (IRUTHAYARAJAN; BASKAR, 2009) o controle PID é uma abordagem de controle muito difundida e eficiente para muitos problemas de controle quando bem sintonizado.

O PID em tempo contínuo é composto por três ações (proporcional, integral e derivativo), tal que:

$$U(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) E(s) \quad (1.1)$$

onde s é a variável de Laplace, K_p é o ganho proporcional, T_i é a constante de tempo integral (segundos), T_d é a constante de tempo derivativo (segundos), $E(s)$ é o sinal de erro, $Y(s)$ é a saída e $U(s)$ é o sinal de controle.

O desafio é determinar valores apropriados de K_p , T_i e T_d que tendem a diminuir o erro e aumentar o tempo de resposta quando utilizados juntos.

Desafio, geralmente mais complexo, é determinar valores para as ações quando há múltiplas entradas e múltiplas saídas, como é o caso do PID multivariável.

1.2.1 Modelo do Controlador Digital

Num modelo de controle digital, a integração ao longo do tempo é substituída pelo somatório de um conjunto discreto finito de instantes, assim, o modelo do controlador digital aproxima-se de um modelo de um controlador em tempo contínuo.

De acordo com Chen (2006), são vários os métodos de discretização de um controlador: Euler em Avanço (do inglês, *Euler's forward method*) dado pela equação (1.2), Euler em Atraso (do inglês, *Euler's backward method*) regido pela equação (1.3), e aproximação de Tustin, também conhecido como Integração Trapezoidal dado pela equação (1.4). Neste caso,

$$s = \frac{z - 1}{T} \quad (1.2)$$

$$s = \frac{z + 1}{zT} \quad (1.3)$$

$$s = \frac{2}{T} \frac{z - 1}{z + 1} \quad (1.4)$$

onde z é a variável complexa e T é o período de amostragem.

Para a discretização do controlador PID, será utilizada a aproximação de Tustin para o termo integral e *backward* para o termo derivativo, conforme mostrado na equação (1.7).

Adotando-se que K_i e K_d são ganhos integrativos e derivativos do controlador PID contínuo representados respectivamente pelas equações (1.5) e (1.6), a equação de controle digital discretizada pela aproximação de Tustin para o termo integral e *backward* para o termo derivativo é representada pela equação (1.7), tal que

$$K_i = \frac{K_p}{T_i} \quad (1.5)$$

$$K_d = K_p T_d \quad (1.6)$$

$$u(k) = u(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \quad (1.7)$$

onde $u(k-1)$ é o valor da ação integral no instante anterior, $e(k)$ é o erro no instante atual, $e(k-1)$ é o erro no instante anterior, $e(k-2)$ é o erro dois passos atrasados, q_0 representa a primeira parcela discretizada do controlador indicado pela equação (1.8), q_1 é a segunda parcela discretizada do

controlador indicado pela equação (1.9) e $q2$ é a parcela derivativa discretizada do controlador, indicado pela equação (1.10), tal que

$$q0 = Kp + \frac{TK_i}{2} + \frac{K_d}{T} \quad (1.8)$$

$$q1 = Kp + \frac{TK_i}{2} - \frac{2K_d}{T} \quad (1.9)$$

$$q2 = \frac{K_d}{T} \quad (1.10)$$

1.2.2 Controle PID Multivariável Acoplado

De acordo com Ram (2015) a estrutura de um controlador PID para sistemas MIMO geralmente é mais complexa quando comparada a estrutura adotada para um sistema SISO (*Single Input, Single Output*) devido as interações entre as variáveis de saída e de entrada. Assim, para controle MIMO têm-se duas variantes: (i) acoplado ou centralizado e (ii) desacoplado ou descentralizado.

O PI desacoplado para um sistema MIMO, como apresentado na FIGURA 1.2, pode ser dimensionado considerando-se vários sistemas SISO paralelos, já que cada entrada possui um único controlador PI e as variáveis de erro são independentes, ou seja, não afetam outros controladores.

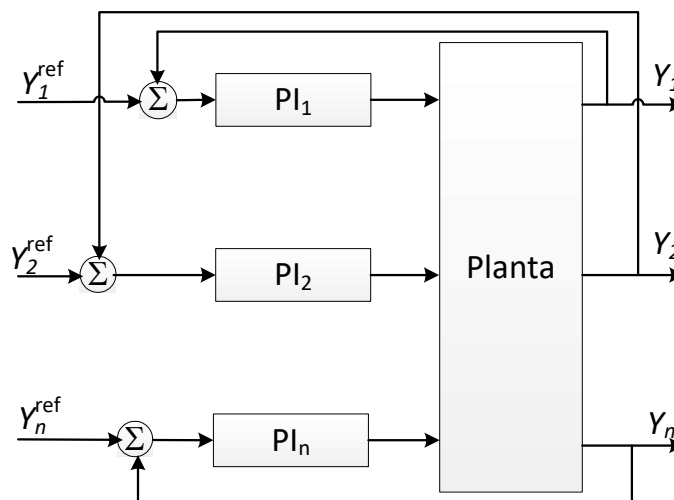


FIGURA 1.2 - CONTROLE PI DESACOPLADO
FONTE: DO AUTOR

Por outro lado, no controlador PI MIMO acoplado (proposto neste trabalho), como exibido na FIGURA 1.3, as variáveis de erro de uma entrada afetam o controlador de outra entrada, onde os ganhos dos controladores (denominados de indivíduos numa otimização) são representados por: $PI_1(k_{11}, ti_1)$, $PI_2(k_{22}, ti_2)$, $PI_n(k_{n3}, ti_3)$, k_{12} , k_{1n} , k_{21} , k_{2n} , k_{n1} , k_{n2} , onde k_{ij} são os ganhos proporcionais do controlador, com i e j sendo as dimensões da matriz e ti_i é o tempo integral.

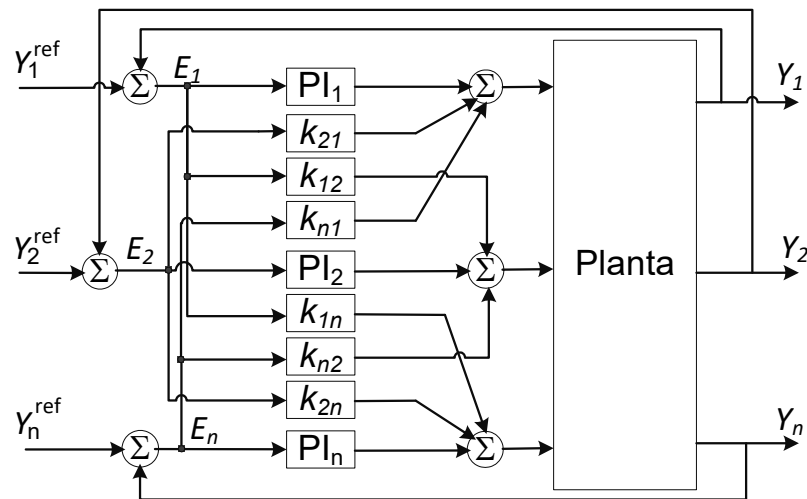


FIGURA 1.3 - CONTROLE PI ACOPLADO
FONTE: DO AUTOR

Um controlador acoplado, conforme comentado por GARRIDO (2012), possui duas abordagens: uma rede desacoplada combinada com uma matriz diagonal do controlador acoplado ou uma estratégia do controlador acoplado puro.

O controlador PI (Proporcional e Integral) acoplado puro é representado por uma matriz apresentada na equação (1.11), onde

$$G_c = \begin{bmatrix} PI_1 & k_{12} & k_{1n} \\ k_{21} & PI_2 & k_{2n} \\ k_{n1} & k_{n2} & PI_n \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \\ E_n \end{bmatrix} \quad (1.11)$$

2 ALGORITMOS EVOLUTIVOS

Os Algoritmos Evolutivos (EAs), que de acordo com Dan (DAN, 2013) refere-se a paradigmas que envolvem soluções de problema com varias iterações - que é representado na natureza por gerações -, tem sua inspiração na Teoria da Evolução e Genética de Darwin (Teoria da Seleção Natural), no qual consiste que os organismos mais adaptados ao ambiente têm maiores chances de sobrevivência, deixando assim, um número maior de descendentes que herdam essas características dos antecessores (pais). A natureza insere um material genético diferente por meio de um processo conhecido como mutação, alterando-se os a genética dos filhos.

Com base nesta teoria, os EAs, utilizados como técnicas de busca e otimização (SOARES, 1997), podem ser modelados por algumas linhas simples de código. Por possuir conceitos biológicos, são classificados como técnicas de Computação Bioinspiradas.

2.1 CARACTERÍSTICAS DOS ALGORITMOS EVOLUTIVOS

Segundo Dan (2013), o EA é indicado para problemas que possuem grande complexidade.

Os EAs têm relação com população de indivíduos (conjunto de soluções), seleção natural (escolha de melhores soluções, ou seja, indivíduos mais aptos ao meio levando em consideração o *fitness*) e *fitness* (valor que classifica os indivíduos candidatos, ou seja, mede o quão adequada é uma determinada solução) (GABRIEL; DELBEM, 2014).

As melhores soluções são selecionadas baseadas no *fitness*, gerando assim, um grupo de população onde serão aplicados operadores genéticos de recombinação (cruzamento) e/ou mutação. A FIGURA 2.1 representa simplificada o procedimento de otimização a lógica que segue o algoritmo evolutivo clássico.

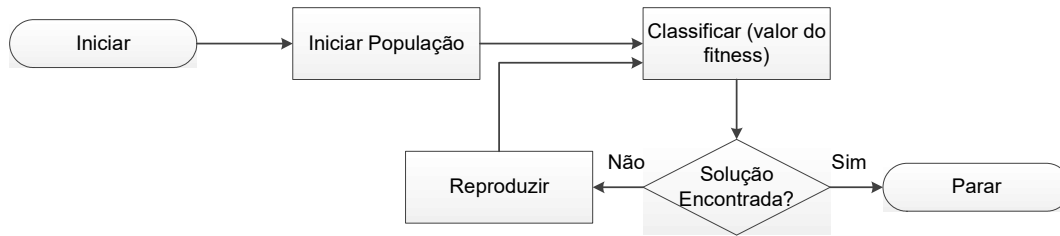


FIGURA 2.1 - CICLO DE OTIMIZAÇÃO DE UM ALGORITMO EVOLUTIVO
 FONTE: O AUTOR (2014)

A mutação dá origem a variabilidade, causando modificações na sequência de base do DNA, podendo favorecer o organismo ou mesmo ser letal (GABRIEL; DELBEM, 2014). A recombinação genética (cruzamento, também denominado de *crossover*) é a combinação entre dois indivíduos, denominados de pais. Os indivíduos resultantes desse cruzamento são identificados como filhos.

A geração é representada por uma iteração do EA, no qual acontece recombinação e/ou mutação gerando descendentes e aumentando a população. Sendo assim, faz-se necessário um mecanismo que controla o tamanho da população, denominado de restrição.

2.2 RESTRIÇÕES

Muitos problemas de otimização reais possuem restrições, podendo ser restrições lineares ou não-lineares. As restrições não-lineares podem formar um espaço de busca complexo. Para não gerar um espaço de busca de complexidade muito elevada, deverá, de acordo com Oliveira (OLIVEIRA, 2014), ser empregadas restrições para manipulação de soluções não viáveis, como: modificar o conjunto para não violar as restrições, e para aquelas soluções que violarem as restrições, deverá ser aplicado penalizações, reduzindo assim a aptidão do indivíduo.

Há várias formas de penalizar uma solução que violou as restrições, como mencionado por Dan (2013). No entanto, a forma de penalidade mais utilizada é por ranqueamento (*ranking*), onde a pior solução viável é melhor que a melhor solução não viável (OLIVEIRA, 2014).

2.3 ALGORITMO DE EVOLUÇÃO DIFERENCIAL

O algoritmo de DE (Evolução Diferencial, do inglês *Differential Evolution*) foi proposto por Storn e Price (STORN; PRICE, 1995). A DE foi concebida com o intuito de perturbar o vetor de população por meio de um vetor de diferenças, teve sua origem no otimizador de método estocástico de combinação baseado em população global, denominado por Recozimento Genético (do inglês, *Genetic Annealing*), este proposto por Price (PRICE, 1994). Porém diferente do seu antecessor, o DE é um otimizador que se baseia em números e não em combinações, sendo assim, aplicável a funções não-lineares e não-diferenciáveis no espaço contínuo. Além disso, é mais simples de implementar do que muitos EAs (DAS, 2011).

O algoritmo DE por ser uma ramificação do algoritmo evolutivo, apresenta mutação e cruzamento, porém sua mutação está baseada na diferenciação entre dois vetores da população selecionados aleatoriamente que perturbam a solução (ZHU *et al.*, 2013), não encontrada (pelo menos até agora) em nenhum outro processo natural.

De acordo com Brest *et al.* (BREST *et al.*, 2000), o DE é composto por três parâmetros de controle: constante de mutação ou peso aplicado ao vetor de diferença (F), constante de *crossover* (Cr) e tamanho da população (NP). Além desses 3 parâmetros tem-se, número de dimensões ou variáveis (D) e número máximo de gerações (gen_{max}). Seu problema está centrado em encontrar um vetor x (vetor de decisão) que otimize a função objetivo ($f(x)$).

Alguns critérios de acordo com Storn (1996 citado por CHAKRABORT, 2008) são levados em consideração quanto a escolha de valores de F , Cr e NP :

- a. F deverá estar no intervalo $[0.5, 1]$. Deve-se ressaltar que quanto maior o tamanho da população, menor deverá ser o valor de F ;
- b. Cr deverá estar no intervalo $[0, 1]$;
- c. NP é aconselhável que seja $10 D$.

A DE pode ser categorizada como um algoritmo onde deve-se fazer a sintonia de parâmetros (do inglês, *Parameter Tuning*), onde os valores de F , Cr

e NP são valores determinados no início do processo e não são alterados no seu decorrer; e controle de parâmetros (do inglês, *Parameter Control*), onde os parâmetros são configurados durante a execução do processo. Por sua vez, o controle de parâmetros é subdividido em: determinístico, onde os parâmetros são alterados de acordo com uma regra determinística; adaptativo, como é o caso do algoritmo de Evolução Diferencial Adaptativa (ADE), como comenta (ZHU *et al.*, 2013), onde os parâmetros alteram-se dinamicamente com base em um *feedback* de busca; e autoadaptativo, onde os parâmetros a sofrerem modificação são codificados nos indivíduos e sofrem alteração dos operadores genéticos, como é o caso do DESAP (do inglês, *Differential Evolution with Self-Adapting Populations*), onde o fator de adaptação encontra-se no NP .

Alguns são os exemplos de algoritmos ADE, como é o caso do: FADE (do inglês, *Fuzzy Adaptive DE*), onde F e Cr são ajustados dinamicamente pelo sistema da base do conhecimento usando uma abordagem de sistemas nebulosos (*fuzzy*); do JADE, onde F e Cr são gerados independentemente em cada geração de acordo com uma distribuição normal e uma determinada distribuição de *Cauchy*; do SaDE (do inglês, *Self-adaptive DE*), onde F e Cr são gerados aleatoriamente de acordo com a distribuição normal em cada geração para cada indivíduo; do jDE (do inglês, *j-DE*), onde NP é reduzido durante a otimização do processo e F e Cr são ajustados por meio de dois argumentos aleatórios.

2.3.1 Estratégias do Algoritmo de Evolução Diferencial

Há várias estratégias que são aplicadas aos DEs modificando-se apenas o vetor de diferença. A TABELA 2.1 representa essas estratégias indicadas por DE/x/y/z, onde DE indica o tipo do algoritmo (evolução diferencial), x é o tipo de vetor que será perturbado e sofrerá a mutação, denominado como vetor base - pode ser aleatório (*rand*) ou o melhor (*best*), ou seja, pode ser aleatório ou o vetor com melhor custo da população, respectivamente, y é a quantidade de vetor de diferença considerando a

perturbação em x , e z é o tipo de cruzamento, podendo ser binominal (*bin*) ou exponencial (*exp*).

TABELA 2.1 - TABELA DE ESTRATÉGIAS DA DE

Estratégia	Mecanismo
1	DE/ <i>best</i> /1/ <i>exp</i>
2	DE/ <i>rand</i> /1/ <i>exp</i>
3	DE/ <i>rand-to-best</i> /1/ <i>exp</i>
4	DE/ <i>best</i> /2/ <i>exp</i>
5	DE/ <i>rand</i> /2/ <i>exp</i>
6	DE/ <i>best</i> /1/ <i>bin</i>
7	DE/ <i>rand</i> /1/ <i>bin</i>
8	DE/ <i>rand-to-best</i> /1/ <i>bin</i>
9	DE/ <i>best</i> /2/ <i>bin</i>
10	DE/ <i>rand</i> /2/ <i>bin</i>

FONTE: DO AUTOR (2016)

Para melhorar a diversidade em problemas onde a população é muito grande, pode-se utilizar duas diferenças ponderadas para perturbar o vetor base, ou seja, são escolhidos cinco vetores distintos baseados em distribuição uniforme.

2.3.2 Algoritmo de Evolução Diferencial Clássico (DE)

O algoritmo clássico do DE é similar ao algoritmo genético como pode ser analisado pela FIGURA 2.2, porém mais rápido que seu antecessor.

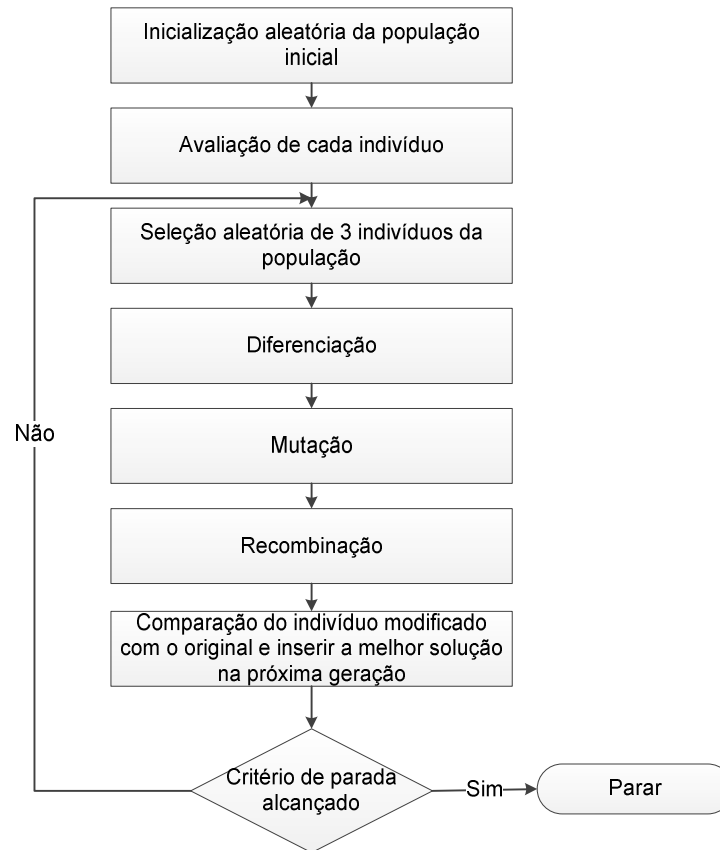


FIGURA 2.2 - ALGORITMO DE EVOLUÇ O DIFERENCIAL
 FONTE: O AUTOR (2014)

De forma extensa, a FIGURA 2.2 pode ser interpretada pelos passos a seguir adotando que a busca do ponto de otimizaç o global   um par metro real no espaço R^D :

- i. A populaç o inicial deve estar dentro dos limites inferiores e superiores de cada vari vel de projeto e inicializada de forma aleat ria, como mostra a equaç o (2.1):

$$x_{i,g} = x_{i,L} + rand_{g,i}[0,1] (x_{i,U} - x_{i,L}) \quad (2.1)$$

onde i   o indiv duo, g   a geraç o, $x_{i,L}$ s o os limites inferiores das vari veis de projeto e $x_{i,U}$ s o os limites superiores das vari veis de projeto e $rand_{g,i}[0,1]$   o n mero aleat rio gerado com distribuiç o uniforme no intervalo 0 e 1;

- ii. Em seguida, a populaç o inicial gerada deve ser avaliada por uma funç o objetivo;

- iii. Depois de avaliada, a população deve sofrer mutação (estratégias do DE), utilizando um vetor-diferença e resultando no vetor de perturbação. A equação (2.2) apresenta o cálculo do vetor de perturbação para a estratégia clássica *rand/1/bin*:

$$v_{i,g} = x_{r_1^i,g} + F(x_{r_2^i,g} - x_{r_3^i,g}) \quad (2.2)$$

onde $x_{r_1^i}$, $x_{r_2^i}$ e $x_{r_3^i}$ são vetores selecionados aleatoriamente da população atual, os índices r_1^i , r_2^i e r_3^i são inteiros escolhidos aleatoriamente entre $[1, NP]$, F é fator de ponderação, ou peso aplicado à diferenciação e $v_{i,g}$ é o vetor de perturbação;

O processo de obtenção do vetor de perturbação descrito pode ser observado no esquema no espaço 2D ilustrado na FIGURA 2.3.

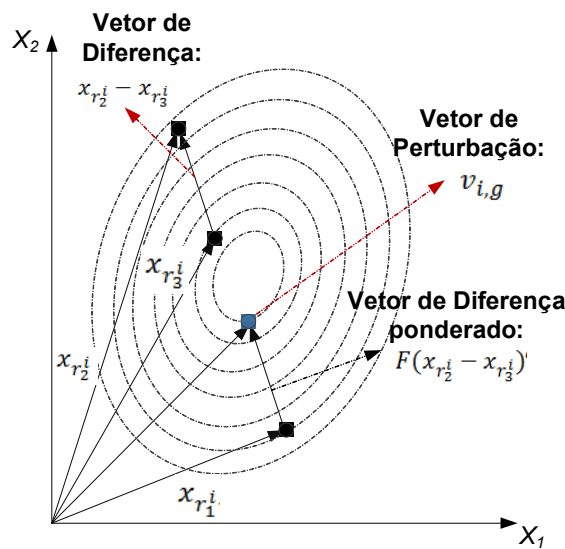


FIGURA 2.3 - ESQUEMA NO ESPAÇO 2-D DO VETOR DE PERTURBAÇÃO
FONTE: DO AUTOR (2015)

- iv. Após sofrer mutação, deve ser aplicado o cruzamento binomial da população para resultar em uma nova geração diversificada, como mostra a equação (2.3);

$$u_{i,g}^j = \begin{cases} v_{i,g}^j & \text{se } \text{rand}_j(0,1) \leq Cr \text{ ou } j = j_{rand} \\ x_{i,g}^j & \text{caso contrário} \end{cases} \quad (2.3)$$

onde Cr é o coeficiente de cruzamento, e $j = 1, 2, \dots, D$.

- v. E por fim, a próxima etapa é a seleção para a nova geração. No DE toda a população - tanto pais, como filhos - compete entre si, ou seja, aquele que tiver melhor valor da função objetivo é selecionado para a próxima geração, como exibido na equação (2.4):

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{se } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{caso contrário} \end{cases} \quad (2.4)$$

onde $f(x)$ é a função objetivo a ser minimizada.

A rotina i. até a rotina v. deve ser repetida até atingir o critério de parada.

De acordo com Das (DAS, 2011) o critério de parada pode ser: 1) fixado por um número de iterações (gen_{max}); 2) quando o melhor *fitness* da população não altera sensivelmente ao fim de sucessivas iterações; e 3) atingir um valor de função objetivo pré-definido.

2.3.3 Algoritmo CMAES (*Covariance Matrix Adaptation* Evolution Strategy)

O CMAES é um algoritmo proposto por Hansen e Ostermeier (HANSEN; OSTERMEIER, 1996) usando como principal mecanismo a adaptação da matriz de covariância (CMA, do inglês, *Covariance Matrix Adaptation*), que tem por finalidade ajustar o tamanho do passo da mutação. O CMAES tem como base a distribuição normal, e o seu objetivo é ser atualizado de maneira acumulada. A equação (2.5) representa o cálculo do indivíduo no CMAES

$$x_{i,g+1} = N(m_g, \sigma_g^2 C_g) \sim m_g + \sigma_g B_g D z_g N(0, I) \sim m_g + \sigma_g N(0, C_g) \quad (2.5)$$

onde, g é a iteração atual, m_g é a média calculada pela equação (2.6), de acordo com Araújo *et al.* (ARAÚJO *et al.*, 2013), C_g é a matriz de covariância, σ_g é o passo de C_g atualizada pela equação (2.7), $B_g \in R^D$ é a matriz ortogonal dos autovetores de C_g , $D z_g \in R^D$ é a matriz diagonal das raízes quadradas dos autovalores de C_g , $N(0, I)$ é a distribuição normal com média zero e matriz de covariância unitária

$$m_g = \sum_{i=1}^{NP} r_i w_{i,g} \quad (2.6)$$

onde, NP é o numero da população, r_i é o valor do peso para a recombinação, no qual $\sum_{i=1}^{NP} r_i = 1, r_i > 0, i = 1, \dots, NP$ e $w_{i,g}$ é o melhor individuo da população,

$$\sigma_{g+1} = \sigma_g \exp \left[\frac{c_\sigma}{d} \left(\frac{\|P_{\sigma,g+1}\|}{E\|N(0,I)\|} - 1 \right) \right] \quad (2.7)$$

em que

$$d = \max \left(1, \frac{3NP}{D+10} \right), \quad c_\sigma = \left(\frac{4}{D+4} \right), \quad (2.8)$$

$$E\|N(0,I)\| = \frac{\sqrt{2}\Gamma(\frac{D+1}{2})}{\Gamma(\frac{D}{2})} \approx \sqrt{D} \left(1 + \frac{1}{4D} + \frac{1}{21D^2} \right)$$

onde D é o numero de variáveis de decisão e $P_{\sigma,g} \in R^D$ é o caminho da evolução conjugada calculado pela equação (2.9).

$$P_{\sigma,g+1} = (1 - c_\sigma)P_{\sigma,g} + \sqrt{c_\sigma(2 - c_\sigma)}B_g(Dz_g)^{-1}(B_g)^{-1}\frac{\sqrt{NP}}{\sigma_g}(m_{g+2} - m_g) \quad (2.9)$$

onde $P_{c,(0)} = P_{\sigma,(0)} = 0$.

2.3.4 Algoritmo CODE (*Composite Differential Evolution*)

Este algoritmo, proposto por Wang *et al.* (2011), compreende na combinação de três vetores de estratégias diferentes (DE/*rand/1/bin* (equação (2.10)), DE/*rand/2/bin* (equação (2.11)) e DE/*current-to-best/1* (equação (2.12))) com três controles de parâmetros fixos para cada estratégia, como exibido na

TABELA 1.

$$v_{i,g} = x_{r_1^i,g} + F(x_{r_2^i,g} - x_{r_3^i,g}) \quad (2.10)$$

$$v_{i,g} = x_{r_1^i,g} + F(x_{r_2^i,g} - x_{r_3^i,g}) + F(x_{r_4^i,g} - x_{r_5^i,g}) \quad (2.11)$$

$$v_{i,g} = x_{i,g} + F(x_{best,g} - x_{i,g}) + F(x_{r_1^i,g} - x_{r_2^i,g}) \quad (2.12)$$

TABELA 1 - CONTROLE DE PARAMETROS PARA ESTRATEGIAS DO CODE

F	Cr
1	0,1
1	0,9
0,8	0,2

FONTE: WANG (2011)

2.3.5 Algoritmo EPSDE (*Ensemble of Mutation and Crossover Strategies and Parameters in Differential Evolution*)

Este é um algoritmo proposto por Mallipeddi *et al.* (2011), consiste num conjunto de estratégias de mutação e cruzamento de técnicas distintas e conjunto de valores para cada parâmetro de controle paralelos competindo entre si para a produção de melhores filhos.

O conjunto de estratégias usadas é: DE/*best*/2 (equação (2.13)) DE/*rand*/1/*bin* (equação (2.14)) e DE/*current-to-rand*/1 (equação (2.15)), tal que.

$$v_{i,g} = x_{best} + F(x_{r_1,g}^i \quad x_{r_2,g}^i) + F(x_{r_3,g}^i \quad x_{r_4,g}^i) \quad (2.13)$$

$$v_{i,g} = x_{r_1,g}^i + F(x_{r_2,g}^i \quad x_{r_3,g}^i) \quad (2.14)$$

$$v_{i,g} = x_{i,g} + F(x_{r_1,g}^i \quad x_{i,g}) + F(x_{r_2,g}^i \quad x_{r_3,g}^i) \quad (2.15)$$

2.3.6 Algoritmo JADE (*Adaptive Differential Evolution with Optional External Archive*)

Esse algoritmo foi desenvolvido por Zhang e Sanderson (2009) e depende de uma estratégia DE/*current-to-best*/1, onde usa o melhor vetor da iteração atual, como exemplifica a equação (2.16), tal que

$$v_{i,g} = x_{i,g} + F_i(x_{best,g} - x_{i,g}) + F_i(x_{r_2,g} - x_{r_3,g}) \quad (2.16)$$

onde $x_{best,g}$ é o vetor com o melhor *fitness* entre os indivíduos da iteração atual.

Os valores de constante de mutação (equação (2.17)) e cruzamento (equação (2.18)) são obtidos por meio de uma distribuição normal (Gaussiana) e uma distribuição de Cauchy. Neste caso tem-se

$$F_i = randc_i(\mu_F; 0,1) \quad (2.17)$$

$$Cr_i = randn_i(\mu_{Cr}; 0,1) \quad (2.18)$$

onde $randc_i$ e $randn_i$ são, respectivamente, a distribuição de Cauchy e a distribuição normal, e μ_F e μ_c são as médias calculadas pelas equações (2.19) e (2.20) respectivamente.

$$\mu_F = (1 - c) \mu_F + c \text{ mean}_L(S_F) \quad (2.19)$$

$$\mu_{Cr} = (1 - c) \mu_{Cr} + c \text{ mean}_A(S_{Cr}) \quad (2.20)$$

onde c é uma constante positiva no intervalo de (0,1), S_F e S_{Cr} são as taxas de todas as mutações/cruzamentos que tiveram sucesso, mean_A e mean_L é media aritmética atual e média Lehmer (calculado de acordo com a equação (2.21)), respectivamente, tal que

$$\text{mean}_L(S_F) = \frac{\sum_{i=1}^{|S_F|} F_i^2}{\sum_{i=1}^{|S_F|} F_i} \quad (2.21)$$

2.3.7 Algoritmo jDE (*j-Differential Evolution*)

Neste algoritmo, proposto por Brest *et al.* (BREST *et al.*, 2006, citado por ZHU *et al.*, 2013), baseado no clássico DE/*rand/1* onde o tamanho da população (NP) é fixa durante o processo de otimização e os parâmetros de controle F e Cr são associados a cada indivíduos e ajustados por meio de dois outros argumentos: τ_1 responsável pela constante de mutação, como mostra a equação (2.22) e τ_2 responsável pela constante de cruzamento, como mostra a equação (2.23).

Por alterar os parâmetros F e Cr , este método adota uma abordagem modificada de mutação, cruzamento e seleção do novo vetor. Neste caso, tem-se

$$F_{i,g+1} = \begin{cases} F_{L,g} + rand_1 & F_{U,g}, & \text{if } rand_2 < \tau_1 \\ F_{i,g}, & & \text{caso contrário} \end{cases} \quad (2.22)$$

$$Cr_{i,g+1} = \begin{cases} rand_3, & \text{if } rand_4 < \tau_2 \\ F_{i,g} & \text{caso contrário} \end{cases} \quad (2.23)$$

onde $rand_k, k \in \{1, 2, 3, 4\}$ são valores distribuídos aleatoriamente no intervalo $[0,1]$, $F_{L,g}$ e $F_{U,g}$ são valores mínimo e máximo, respectivamente, do parâmetro da constante de mutação, $F_{i,g+1}$ e $Cr_{i,g+1}$ são os novos valores dos parâmetros de mutação e cruzamento.

2.3.8 Algoritmo SaDE (Self-adaptive Differential Evolution)

Neste algoritmo, proposto por Qin *et al.* (2009), é empregado quatro variações da estratégia de mutação do clássico DE: DE/*rand/1* (equação (2.24)); DE/*current-to-best/2* (equação (2.25)); DE/*rand/2* (equação (2.26)); DE/*current-to-rand/1* (equação(2.27)); onde o Cr é codificado no indivíduo e o F é geral. As questões neste caso são as seguintes

$$v_{i,g} = x_{r_1^i,g} + F(x_{r_2^i,g} - x_{r_3^i,g}) \quad (2.24)$$

$$v_{i,g} = x_{i,g} + F(x_{best,g} - x_{i,g}) + F(x_{r_1^i,g} - x_{r_2^i,g}) + F(x_{r_3^i,g} - x_{r_4^i,g}) \quad (2.25)$$

$$v_{i,g} = x_{r_1^i,g} + F(x_{r_2^i,g} - x_{r_3^i,g}) + F(x_{r_4^i,g} - x_{r_5^i,g}) \quad (2.26)$$

$$v_{i,g} = x_{i,g} + F(x_{r_1^i,g} - x_{r_2^i,g}) \quad (2.27)$$

onde $x_{best,g}$ é o vetor com o melhor *fitness* entre os indivíduos da iteração atual.

O parâmetro de F é calculado de acordo com uma distribuição normal apresentado na equação (2.28) e Cr é inicialmente fixado em 0,5 e alterado no período de aprendizado pela distribuição normal com desvio de 0,1. Depois do período de aprendizado o Cr_i é calculado usando a mediana do conjunto dos Cr das melhores soluções, conforme mostrado na equação (2.29), tal que

$$F_i = randn_i(0,5; 0,3) \quad (2.28)$$

$$Cr_i = randn_i(Crm; 0,1) \quad (2.29)$$

onde $randn_i$ é a distribuição normal para cada iteração e Crm é a mediana do Cr depois do período de aprendizado.

2.4 OUTROS ALGORITMOS BASEADOS EM POPULAÇÃO

Alguns algoritmos baseados em população foram selecionados para a comparação com o algoritmo DE clássico: PSO (*Particle Swarm Optimization*), MVO (*Multi Verse Optimizer*), GWO (*Grey Wolf Optimizer*), ALO (*Anti Lion Optimizer*) e DA (*Dragonfly Algorithm*).

2.4.1 Algoritmo PSO (*Particle Swarm Optimization*)

O PSO (otimização por enxame de partículas, do inglês, *Particle Swarm Optimization*), desenvolvido por Kennedy *et al.* (1995), tem como base os animais que vivem coletivamente, como por exemplo, peixes e pássaros. O PSO é um algoritmo da classe denominada de inteligência de enxames (*swarm intelligence*). Seu pseudo-código é apresentado na FIGURA 2.4.

```

Iniciar vetor de posição ( $x$ )
Iniciar o vetor de velocidade ( $v$ )
Obter valores iniciais para  $gbest$  e  $pbest$ 
enquanto o fim do critério não for satisfeito
    Calcular o  $fitness$ 
    se  $fitness(x)$  melhor que  $fitness(gbest)$ 
         $gbest_d = x_{id}$ 
    fim do se
    se  $fitness(x)$  melhor que  $fitness(pbest)$ 
         $pbest_d = x_{id}$ 
    fim do se
    Atualizar a posição usando a equação de velocidade
    Atualizar a posição usando a equação de posição
fim do enquanto
 $gbest_d = mel \text{ or } solução$ 

```

FIGURA 2.4 - PSEUDO-CÓDIGO DO ALGORITMO PSO
 FONTE: COELHO, 2009

De acordo com Coelho (2013), o PSO é fundamentado nas analogias comportamento cognitivo (*local best knowledge*) e comportamento social (*global best knowledge*), apresentado na equação responsável pela velocidade (equação (2.30)), e suas partículas “voam” em torno do espaço de busca.

$$v_{id} = w \cdot v_{id} + c_1 \cdot ud \cdot (pbest_{id} - x_{id}) + c_2 \cdot UD \cdot (gbest_d - x_{id}) \quad (2.30)$$

onde w é a ponderação de inércia, c_1 é a atração responsável pela parcela cognitiva, em que $pbest \in [0; 2,05]$, c_2 é a atração responsável pela parcela social, em que $gbest \in [0; 2,05]$, ud e UD são números gerados com distribuição uniforme, $pbest_i$ é a posição prévia com melhor *fitness*, $gbest_i$ é o número de partículas vizinhas da posição que resulta no melhor *fitness*.

Por fim, a posição é calculada de acordo com a equação (2.31) dada por

$$x_{id} = x_{id} + t \cdot v_{id} \quad (2.31)$$

onde t é o período de tempo.

2.4.2 Algoritmo MVO (*Multi Verse Optimizer*)

O MVO (otimização de múltiplos universos, do inglês, *Multi Verse Optimizer*) foi proposto por Mirjalili e Mirjalili (2016) e é inspirado na teoria da Física sobre o multi-universo, baseado em três conceitos da cosmologia: buraco branco, buraco preto, e buraco de minhoca. O buraco branco e o buraco negro são utilizados, na ordem, para explorar o espaço de procura enquanto que o buraco de minhoca assiste a exploração feita pelo MVO. Cada solução é representada por um universo e cada variável da solução é um objeto deste universo.

Na FIGURA 2.6 mostra o um pseudo-código do algoritmo MVO.

```

Criar universos aleatórios (U)
Iniciar:
 $l_b$ =variável mínima,  $u_b$ =variável máxima,  $Taxa_{viagem}$ =um coeficiente

Mel orUniverso
SU=Universos aleatórios
NI=fitness dos universos
r1,r2,r3 e r4 = números aleatórios entre 0 e 1
enquanto o fim do critério não for satisfeito
    Avaliar o fitness de todos os universos
para cada universo indexado por i
    Atualiza a  $Taxa_{viagem}$ 
    buraco_negro = i
    para cada objeto indexado por j
        r1=random([0, 1])
        se (r1<NI(Ui))
            buraco_branco = RouletteWheelSelection(-NI)
            U(buraco_negro,j) = SU(buraco_branco,j)
        Fim do se
        r2=random([0, 1])
        se (r2<probabilidade de existência de buraco de minhoca)
            r3= random([0, 1])
            r4=random([0, 1])
            se (r3<0,5)
                U(i,j)=Melhor_Univero(j)+  $Taxa_{viagem} * ((ub(j)-$ 
                lb(j))*r4+lb(j)
            senão
                U(i,j)=Melhor_Univero(j)-  $Taxa_{viagem} * ((ub(j)-$ 
                lb(j))*r4+lb(j)
            fim do se
        fim do se
    fim do para
fim do para
fim do enquanto

```

FIGURA 2.5 - PSEUDO-CÓDIGO DO ALGORITMO MVO
 FONTE: DO AUTOR (2016)

A $Taxa_{viagem}$ é calculada de acordo com a equação (2.32).

$$Taxa_{viagem} = 1 - \frac{l^{1/p}}{L^{1/p}} \quad (2.32)$$

onde l indica a iteração corrente, L indica a máxima iteração, e p indica a exploração.

2.4.3 Algoritmo GWO (*Grey Wolf Optimizer*)

O GWO (otimização dos lobos cinzentos, do inglês, *Grey Wolf Optimizer*) proposto por Mirjalili *et al.* (2014) é inspirado no comportamento de lobos cinzentos que seguem uma hierarquia exibida na FIGURA 2.6.

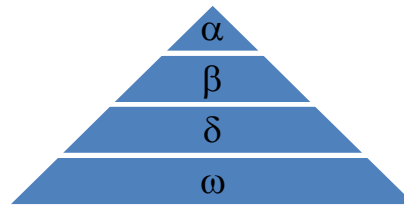


FIGURA 2.6 - HIERARQUIA DE LOBOS CINZENTOS
FONTE: DO AUTOR (2016)

Na FIGURA 2.6, nota-se a presença de alguns parâmetros, em que o α é considerado a primeira melhor solução. A FIGURA 2.7 mostra o um pseudo-código do algoritmo GWO.

```

Iniciar a população de lobos cinzentos  $X_i (i = 1, 2, \dots, n)$ 
Iniciar:
a=parâmetro responsável pela exploração
A e C=parâmetros aleatórios que auxiliam as soluções candidatas
Calcular o fitness de cada agente
 $X_a$ =procura do melhor agente
 $X_g$ =procura do segundo melhor agente
 $X_d$ =procura do terceiro melhor agente
enquanto (t < máximo de iterações)
    para agente da procura
        Atualização da posição do agente da procura corrente
    fim do para
    Atualiza a, A e C
    Calcula o fitness de todos os agentes da procura
    Atualiza  $X_a$ ,  $X_g$  e  $X_d$ 
    Atualiza o contador de iterações, t=t+1
fim do enquanto
Retorna  $X_a$ 

```

FIGURA 2.7 - PSEUDO-CÓDIGO DO ALGORITMO GWO
 FONTE: DO AUTOR (2016)

2.4.4 Algoritmo ALO (*Ant Lion Optimizer*)

ALO (otimização da formiga-leão, do inglês, *Ant Lion Optimizer*) é um algoritmo proposto por Mirjalili (2015) que trata da interação entre a armadilha da formiga-leão e formigas.

A FIGURA 2.8 mostra o pseudo-código do ALO.

Iniciar a primeira população de formigas e formigas-leão aleatoriamente
 Calcular o *fitness* de formigas e formigas-leão
 Procurar a melhor formiga-leão e assumir como elite (determinar ótimo)
enquanto o fim do critério não for satisfeito
 para para toda formiga
 Selecionar uma formiga-leão usando Roulette wheel
 Atualizar c e d
 Criar um caminho aleatório e normalizá-lo (X_i^t)
 Atualizar a posição da formiga
fim do para
 Calcular o *fitnesss* de todas as formigas
 Substituir a formiga-leão que corresponde a formiga que tornou-se
 mais apta
 Atualizar elite se formiga-leão tornar-se mais apto que a elite
fim do enquanto
Retorna elite

FIGURA 2.8 - PSEUDO-CÓDIGO DO ALGORITMO ALO
 FONTE: DO AUTOR (2016)

Os valores de c^t e d^t descritos no pseudo-código é calculado de acordo com a equação (2.33).

$$c^t = \frac{c^t}{l}, d^t = \frac{d^t}{l} \quad (2.33)$$

onde c^t é o mínimo de todas as variáveis da t iteração, d é o máximo de todas as variáveis da t iteração, t é a iteração atual, l é a relação indicada por $l = 10^w \frac{t}{T}$, em que, T é o número máximo de iteração, w é a constante definida baseada na iteração atual ($w = 2$ quando $t > 0,1T$, $w = 3$ quando $t > 0,5T$, $w = 4$ quando $t > 0,75T$, $w = 5$ quando $t > 0,9T$, $w = 6$ quando $t > 0,95T$).

O caminho aleatório das formigas é calculado pela equação (2.34) e atualizado pela equação (2.35).

$$X_i^t = \frac{(X_i^t - a_i)x(d_i - c_i^t)}{(d_i^t - a_i)} + c_i \quad (2.34)$$

$$Ant_i^t = \frac{(R_A^t + R_E^t)}{2} \quad (2.35)$$

onde X_i^t é o caminho aleatório das formigas, a_i é o caminho aleatório mínimo da i variável, b_i é o caminho aleatório máximo da i variável, c_i^t é o mínimo da i variável na t iteração e d_i^t indica o máximo da i variável da t iteração, Ant_i^t indica a posição da i formiga na t iteração, R_A^t é o caminho aleatório em torno da formiga-leão selecionado pelo *roulette wheel* na t iteração, R_E^t é o caminho aleatório em torno da elite na t iteração

A substituição da formiga-leão é feita pela equação (2.36).

$$Antlion_j^t = Ant_i^t \text{ se } f(Ant_i^t) > f(Antlion_j^t) \quad (2.36)$$

2.4.5 Algoritmo DA (Dragonfly Algorithm)

O algoritmo DA (algoritmo da libélula, do inglês, *Dragonfly algorithm*) é um algoritmo proposto por Mirjalili (2015) que possui duas versões: para proposta binária (BDA – *binary dragonfly algorithm*) e multi-objetivo (MODA – *multi-objective dragonfly algorithm*).

A FIGURA 2.9 mostra o pseudo-código do MODA.

```

Iniciar população de libélulas  $X_i(i=1,2,\dots,n)$ 
Iniciar o vetor de passos  $\Delta X_i(i=1,2,\dots,n)$ 
Definir o número máximo de segmentos (hyper spheres)
Definir o tamanho do arquivo
enquanto o fim do critério não for satisfeito
    Calcular o valor objetivo de todas as libélulas
    Encontrar a solução não-dominante
    Atualizar o arquivo com as soluções não-dominantes
    se o arquivo estiver completo
        Rodar o mecanismo de manutenção do arquivo para omitir
        um membro do arquivo atual
        Adicionar a nova solução ao arquivo
    fim do se
    se nenhuma nova solução adicionada ao arquivo é local dentro da
    hyper spheres
        Atualizar e reposicionar todos os hyper spheres para cobrir a
        nova solução
    fim do se
    Selecionar a fonte de alimento do arquivo:
     $X^+=\text{SelecionarComida}(\text{arquivo})$ 
    Selecionar um inimigo do arquivo:  $X^-=\text{SelecionarInimigo}(\text{arquivo})$ 
    Atualizar vetor de passos
    Atualizar posição do vetor
    Verificar e corrigir a nova posição baseada nas fronteiras das
    variáveis
fim do enquanto

```

FIGURA 2.9 - PSEUDO-CÓDIGO DO ALGORITMO MODA
 FONTE: DO AUTOR (2016)

2.5 OTIMIZAÇÃO MULTIOBJETIVO

Os problemas podem ser classificados como otimização mono-objetivo (SOP, do inglês, *single-objective optimization*) ou otimização multiobjetivo (MOP, do inglês, *multi-objective optimization*). Os SOPs trabalham com somente um valor ótimo-global, já os MOPs trabalham com vários objetivos simultâneos.

Um problema de otimização multiobjetivo (MOP) tem por finalidade encontrar valores de variáveis que minimizam as várias funções objetivo, otimizando o sistema sem perda de generalidade, como pode ser analisado na equação (2.37), satisfazendo a m inequações e p equações de restrição das funções, representados pelas equações (2.38) e (2.39), respectivamente atendendo as condições de contorno da equação (2.40).

$$\min_x f(x) = \min_x y = [f_1(x), f_2(x), \dots, f_k(x)] \in Y \quad (2.37)$$

$$\text{Sujeito à } g_i(x) \leq 0 \quad i = 1, 2, \dots, m \quad (2.38)$$

$$h_i(x) = 0 \quad i = 1, 2, \dots, p \quad (2.39)$$

$$x_i^L \leq x_i \leq x_i^U \quad i = 1, 2, \dots, D \quad (2.40)$$

onde x é o vetor de decisão ($x = [x_1, \dots, x_n]^T \in X$, em que X é o espaço de parâmetros), k é o número de funções objetivo (se $k > 1$ o sistema é considerável um problema multiobjetivo), $f(x)$ é o vetor objetivo, Y é o espaço objetivo, x_i^L é o limite inferior e x_i^U é o limite superior do vetor de decisão e D é o número da dimensão.

2.5.1 Definição de Pareto-ótimo

Uma única solução não atende a todos os critérios de um sistema MOP, no entanto, é possível obter um conjunto de soluções denominado de conjunto de Pareto-ótimo, que é a mais adequada para todos os objetivos.

Dadas duas soluções $\mathbf{a} \in X$ e $\mathbf{b} \in X$, é dito que \mathbf{a} domina \mathbf{b} (denotado por $\mathbf{a} \succ \mathbf{b}$) se e somente se:

- a. $\forall i \in \{1, \dots, n\}: f_i(\mathbf{a}) \leq f_i(\mathbf{b})$: \mathbf{a} é melhor ou igual a \mathbf{b} em todas as funções objetivo, onde n é o número de funções objetivo;
- b. $\exists j \in \{1, \dots, n\}: f_j(\mathbf{a}) < f_j(\mathbf{b})$: \mathbf{a} é estritamente melhor que \mathbf{b} em pelo menos uma função objetivo.

A FIGURA 2.10 mostra um exemplo de dominância: o ponto B domina a solução representada pelo ponto A.

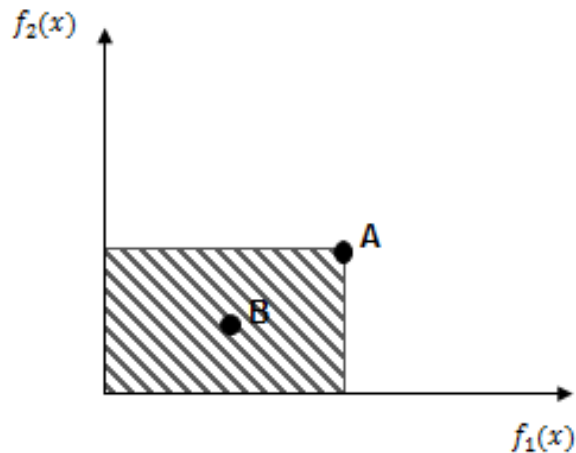


FIGURA 2.10 - SOLUÇÃO DE DOMINÂNCIA
 FONTE: AZUMA, 2011

Todo vetor de decisão que não é dominado por outro vetor de decisão é chamado de não-dominado. De acordo com Azuma (AZUMA, 2011) o conjunto de soluções não-dominadas é chamado de conjunto Pareto-ótimo e representa a solução do problema de minimização. Já a fronteira de Pareto é o conjunto dos valores das funções objetivo encontrados nas soluções do conjunto Pareto-ótimo.

De acordo com AZUMA (2011) existem três metas em otimização multiobjetivo: obter um conjunto de soluções que esteja mais próximo da fronteira de Pareto (meta de toda otimização, seja ela SOP ou MOP e também a utilizada nesta dissertação); obter um conjunto de soluções com maior diversidade (meta específica das otimizações multiobjetivo); e alcançar as metas com a maior eficiência computacional.

Analisando a FIGURA 2.11 que representa a primeira e a segunda meta, pode-se ver que o algoritmo deve contemplar as metas de diversidade e convergência já que são conflitantes.

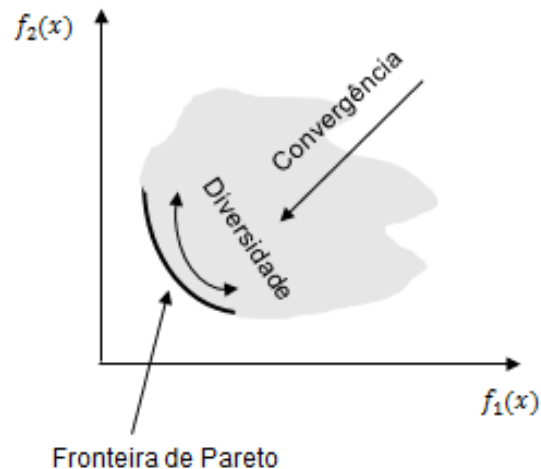


FIGURA 2.11 - METAS DA OTIMIZAÇÃO MULTIOBJETIVO
 FONTE: AZUMA (2011)

2.5.2 Classificação de um Problema de Otimização Multiobjetivo

Um problema multiobjetivo trabalha com o espaço das variáveis e o espaço dos objetivos, podendo ser convertido (utilizando algumas técnicas, como por exemplo, somatório de pesos, métodos de restrições e programação por metas, explicadas por Ticona e Delbem (TICONA; DELBEM, 2008)) em um problema de otimização simples, no qual trabalha apenas com o espaço das variáveis, ou então transformar o problema numa otimização mono-objetivo – combinar vários objetivos em um único objetivo. Assim, faz-se necessário métodos que consigam lidar com as variáveis sem a necessidade de conversão para um modelo mais simples, ou seja, sem perda de informação.

Os problemas multiobjetivos são classificados, de acordo com Sampaio (2011), pelo tipo de decisão que utilizam:

- a. *a-priori*, que é a tomada de decisão antes da busca de soluções do problema, tratando-o como um mono-objetivo, ou seja, combina todos os objetivos em um único objetivo. Por buscar um único ponto ótimo, requer menor esforço computacional, porém não há informações sobre o conjunto de alternativas, necessitando de novas buscas caso seja alterado as

preferencias do objetivo e inclusões de novos aspectos não considerados anteriormente;

- b. *a-posteriori*, que é a tomada de decisão depois da busca de soluções e considera que todos os objetivos possuem a mesma relevância para o problema. Este método por trabalhar com todos os objetivos, requer mais esforço computacional; e
- c. iterativos, que é a tomada de decisão durante a busca de soluções, que guia a busca de conjunto de soluções conflitantes a cada iteração.

De acordo com Meza (2014), vários algoritmos são utilizados para identificar os melhores valores de um MOP, como por exemplo, *genetic algorithms* (GA), *particle swarm optimization* (PSO) e *differential evolution* (DE) (DAS, 2011), pois demonstram uma excelente capacidade de busca para soluções ótimas (LIN, 2015), quando tratado com múltiplos objetivos simultaneamente.

Nesta dissertação o tipo de decisão aplicado foi o *a-priori*, ou seja, tratando o sistema multiobjetivo como um sistema mono-objetivo. Para isso, a soma ponderada dos sinais de erro foi aplicada aos estudos de caso adotados.

3 DESCRIÇÃO DOS ESTUDOS DE CASO

Dois estudos de caso foram abordados, são eles: *Boiler Turbine* e *Quadruple Tank*.

Para validar os algoritmos nos estudos de caso adotados, foram utilizados algoritmos baseados em população para a sintonia do controlador PI (proporcional e integral) do tipo acoplado e discretizado. Para cada estudo foi simulado adotando o algoritmo DE clássico alterando-se os parâmetros de aplicação das operações de mutação e cruzamento. Os resultados obtidos pelo DE foram comparados com outras metaheurísticas de otimização. Neste caso são avaliados os valores mínimo, máximo, média e o desvio padrão em termos da função objetivo adotada.

Para analisar o desempenho do algoritmo DE em comparação a outros algoritmos descritos nesta dissertação, foi utilizado o teste de Wilcoxon-Mann-Whitney U-test. Este é um teste não paramétrico que compara as tendências centrais (medianas) entre duas amostras independentes, mas de tamanhos iguais. Em alguns casos, o teste de Wilcoxon-Mann-Whitney U-test não foi o suficiente para a análise, assim foi necessário outro teste denominado Teste T que usa o *one-sample* t-Test (teste paramétrico de localização do parâmetro quando o desvio padrão é desconhecido, comparando-se as médias) adotando nível de significância $\alpha=5\%$ (probabilidade de rejeitar a hipótese quando ela é verdadeira, adotado pelo comando *ranksum* do matlab).

3.1 TESTE DE WILCOXON-MANN-WHITNEY U-TEST

O teste de Wilcoxon-Mann-Whitney U-test é baseado em *ranks* dos valores obtidos e em hipóteses. Duas são as hipóteses usadas: H_0 (hipótese nula) e H_1 (hipótese alternativa). Este teste serve para analisar se dois grupos pertencem a uma mesma população.

O procedimento para a análise é realizado usando um nível de significância $\alpha=5\%$ (probabilidade de rejeitar a hipótese quando ela é verdadeira) e seguindo as seguintes etapas:

1. Inicialmente a hipótese nula é considerada verdadeira;
2. O teste não paramétrico é aplicado ao conjunto de dados para a decisão da rejeição da hipótese nula;
3. Caso a hipótese nula não seja rejeitada, ela é considerada verdadeira. Caso seja rejeitada, a hipótese alternativa é considerada verdadeira;

Nos dados, se a hipótese for igual a 1 ($h=1$) indica a rejeição da hipótese nula, ou seja, há evidências de diferença entre as amostras. Caso a hipótese seja igual a 0 ($h=0$) houve a falha ao rejeitar a hipótese nula, assim as amostras são parecidas.

Também é apresentado a *p-value* (escalar positivo entre 0 e 1) definida por FACHIN (2014) como uma estatística obtida através do teste não paramétrico aplicado aos dados em análise. Assim, de acordo com Walpole *et al.* (2011, citado por FACHIN, 2014) quanto menor o *p-value* maior é a evidência para a rejeição da hipótese nula, ou seja, quanto menor o *p-value*, maior é a probabilidade das amostras serem diferentes ($h=1$).

3.2 ÍNDICES DE DESEMPENHO

Um índice de desempenho é útil para a análise de desempenho de um controle de sistema. Um dos mais utilizados é o critério de minimização da integral do erro, citados em BHATTACHARYA (2009), tais como: ISE (integral do erro ao quadrado, do inglês, *integral of square of the error*), apresentado na equação (3.1); ITSE (integral do quadrado do erro multiplicado pelo tempo, do inglês, *integral of time multiplied by square error*), regido pela equação (3.2); IAE (integral absoluta do erro, do inglês, *integral of the absolute magnitude of the error*), apresentado na equação (3.3); e ITAE (integral do erro absoluto multiplicado pelo tempo, do inglês, *integral time-weighted absolute error*). As equações que regem os índices mencionados são as seguintes:

$$ISE = \int_0^T e^2(t)dt \quad (3.1)$$

$$ITSE = \int_0^T te^2(t)dt \quad (3.2)$$

$$IAE = \int_0^T |e_1(t)|dt \quad (3.3)$$

$$ITAE = \int_0^T t|e_1(t)|dt \quad (3.4)$$

onde t é o tempo, T é o tempo máximo da simulação e $e(t)$ é o erro da saída em relação ao sinal de referência.

De acordo com FERMINO (2014), o ISE é um índice que penaliza grandes erros e tem uma resposta rápida, porém deixa o sistema com muitas oscilações, enquanto que o ITSE (utilizado neste trabalho como critério de desempenho) é um índice que penaliza erros que persistem por longos períodos de tempo, ou seja, erros em regime permanente. Já o IAE necessita que a simulação seja executada por um grande período de tempo para haver um valor significativo, apresentando uma resposta com sobressinal, mas não oscilatória. E por fim, um sistema que utiliza o ITAE apresenta um pequeno sobressinal e oscilações bem amortecidas.

3.3 FUNÇÃO OBJETIVO

O *ITSE* discretizado, descrito na equação (3.5) é a função a minimizar, ou função objetivo, tal que

$$f = \sum_{i=0}^k t(i) (e_1(i)^2 + e_2(i)^2 + \dots + e_n(i)^2) \quad (3.5)$$

onde f é a função *ITSE* de todos os erros, i é a iteração, k é o tempo de quantidade de amostras, $t(i)$ é o tempo em cada instante, e_1 é o erro relacionado a primeira saída, e_2 é o erro relacionado a segunda saída, e_n é o erro relacionado a última saída e n é o quantidade total das saídas.

De acordo com os valores de sinal de controle, a equação (3.5) é penalizada, gerando assim a função objetivo.

O modelo trabalhado da planta e do controlador foi o discretizado. Assim, todas as equações dos estudos de foram discretizadas de acordo com a aproximação de Euler, descrita na equação (3.6).

$$x(k) = x(k-1) + T f(k) \quad (3.6)$$

onde $x(k)$ é o valor pretendido no instante atual, $x(k-1)$ é o valor atrasado em um passo e $f(k)$ é a função no instante.

3.4 PARÂMETROS DE SIMULAÇÃO

Para os dois estudos de caso foram executados três testes:

- a) o primeiro testando o algoritmo clássico DE alterando-se fator de cruzamento e mutação;
- b) o segundo comparando o algoritmo DE clássico com suas variantes;
- c) o terceiro comparando o algoritmo DE clássico com outros algoritmos baseados em população.

Para todos os testes, foram realizados 50 experimentos com 150 iterações e uma população igual a 120. Para todos os casos, foram realizadas análises da saída e análise Wilcoxon-Mann-Whitney U-test com nível de significância $\alpha=5\%$.

Para o primeiro teste, foram realizados 4 (I, II, III e IV) conjuntos de testes alterando-se os parâmetros do projeto de acordo com a TABELA 3.1 utilizando-se da estratégia 7.

TABELA 3.1 - VALORES DO FATOR DE CRUZAMENTO E MUTAÇÃO UTILIZADOS – SISTEMA BOILER

Estudo	Cruzamento (Cr)	Mutação (F)
I	0,8	0,6
II	0,4	0,6
III	0,4	0,4
IV	0,8	0,8

FONTE: DO AUTOR (2016)

No segundo e terceiro teste, cada algoritmo apresenta uma particularidade de parâmetros a sintonizar inicialmente: para o CMAES foi utilizado um passo de 0,2; para o CODE foi usado $F = [0,1 \ 0,2 \ 0,3]$ e $Cr = [0,7 \ 0,9 \ 0,8]$; para o EPSDE foi utilizado $F = [0,3 \ 0,2 \ 0,1 \ 0,65 \ 0,8 \ 0,9]$ e $Cr = [0,1 \ 0,2 \ 0,3 \ 0,4 \ 0,5 \ 0,6 \ 0,7 \ 0,8 \ 0,9]$; para o JADE foi usado $Fm = 0.4$, $Crm = 0.8$, $p = 0,05$ e $c = 0,1$; para o jDE foi usado $\tau_1 = \tau_2 = 0,1$; para o SADE foi usado uma aprendizagem em de 50, $F = 0.4$ e $Cr = 0.8$; para o PSO foi usado velocidade mínima de 0.2, velocidade máxima de 0.9 e $c_1 = c_2 = 2$ (parcela cognitiva e parcela social); para o MVO foi usado o mínimo e máximo da probabilidade de existência de um buraco de minhoca igual a 0,2 e 1, respectivamente. Para os algoritmos citados na dissertação e que não foram apresentados os parâmetros neste parágrafo, usam parâmetros calculados internamente.

3.5 ESTUDO DE CASO 1 – BOILER AND TURBINE

Primeiro desenvolvimento de Bell and Åström, *Boiler and Turbine* (BELL; ÅSTROM, 1987) é um modelo de terceira ordem não-linear MIMO com rígidas restrições e limites impostos sobre os atuadores, baseado na planta P16/G16 do Sydsvenska Kraft AB em Malmö, Suécia, com caldeira chegando a 160MW de potência.

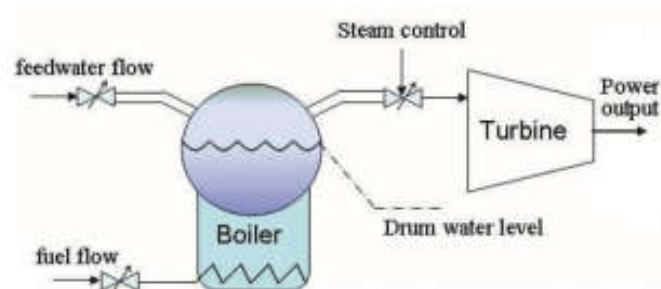


FIGURA 3.1 - ESQUEMA DO BOILER TURBINE
 FONTE:LU; LIN; FEND; WAN (2010)

Com o passar dos anos o modelo original foi modificado para um modelo de baixa ordem (segunda ordem) que representa algumas complexidades dinâmicas associadas com a planta real. Esse sistema dinâmico é modelado pelas equações (3.7), (3.8) e (3.9), tal que:

$$\frac{dp}{dt} = 0,0018u_2p^{\frac{9}{8}} + 0,9u_1 \quad 0,15u_3 \quad (3.7)$$

$$\frac{dP_0}{dt} = (0,073u_2 \quad 0,016)p^{\frac{9}{8}} \quad 0,1P_0 \quad (3.8)$$

$$\frac{dp_f}{dt} = \frac{(141u_3 \quad (1,1u_2 \quad 0,19)p)}{85} \quad (3.9)$$

onde p é a pressão do cilindro (kg/cm^2), P_0 é a potência de saída (MW), e p_f é a densidade do fluido (kg/m^3), u_1 é a válvula de posição do fluxo de combustível, u_2 é a válvula de posição do controle de vapor e u_3 é a válvula de posição do fluxo da água de alimentação.

A limitação da equação (3.10) é imposta sobre a válvula do fluxo de combustível, já a limitação para a válvula do controle de vapor é expressa pela equação (3.11) e a limitação para a válvula do fluxo de água de alimentação é representada pela equação (3.12), e todas as variáveis de posição das válvulas devem estar num intervalo $[0, 1]$ tal que

$$\left| \frac{du_1}{dt} \right| \leq 0,007/s \quad (3.10)$$

$$2/s \leq \frac{du_2}{dt} \leq 0,02/s \quad (3.11)$$

$$\left| \frac{du_3}{dt} \right| \leq 0,05/s \quad (3.12)$$

As entradas do sistema são representadas por u_1 , u_2 e u_3 e as saídas são p , P_0 e X_w (nível da água em metros no cilindro). O nível da água é encontrado utilizando-se a pressão e potência de saída, já que estas são as duas primeiras variáveis de estado, de acordo com a equação (3.13), (3.14) e (3.15).

$$X_w = 0.05(0,13073p_f + 100\alpha_{cs} + \frac{q_e}{9} \quad 67,975) \quad (3.13)$$

$$\alpha_{cs} = \frac{(1 \quad 0,001538p_f)(0,8p \quad 25,6)}{p_f(1.0394 \quad 0.0012304p)} \quad (3.14)$$

$$q_e = (0,8548u_2 \quad 0,147)p + 45,59u_1 \quad 2,514u_3 \quad 2,096 \quad (3.15)$$

onde X_w é o nível da água (m), q_e é a evaporação (kg/s) e α_{cs} é a qualidade do vapor.

Os valores iniciais das variáveis de entrada, controle e saída, utilizados nesta dissertação são mencionados na TABELA 3.2.

TABELA 3.2 - VALORES INICIAIS PARA AS VARIÁVEIS DE ENTRADA, CONTROLE E SAÍDA

Variável	Valor
x_1^0	108
x_2^0	66,65
x_3^0	428
u_1^0	0,34
u_2^0	0,69
u_3^0	0,436
y_1^0	x_1^0
y_2^0	x_2^0
y_3^0	0

FONTE: BELL; ÅSTROM (1987)

O objetivo é controlar a planta para que a pressão e a potência cheguem em 120 e o nível permaneça em 0, ou seja, $y = [120,120,0]$.

Muitas são as técnicas em trabalhos recentes aplicadas ao sistema os de *boiler turbine*, como é o caso da proposta por: Sayed (SAYED *et al.*, 2015), que usa uma estratégia que baseada em Gaussian e Cauchy na mutação denominada HJPSO (do inglês, *hybrid jump particle swarm optimization*) para sintonizar os ganhos do controlador PI (Proportional Integral).; Ghabraei, (GHABRAEI *et al.*, 2015), que usa o RASMC (do inglês, *robust adaptive sliding mode controller*), que é estabilizado pela teoria de Lyapunov.

3.5.1 Resultados DE Clássico

A TABELA 3.3 mostra a mínima e máxima função objetivo e o desvio padrão para os 4 conjuntos de testes realizados.

TABELA 3.3 - VALORES DE MÍNIMA E MÁXIMA FUNÇÃO OBJETIVO E DESVIO PADRÃO PARA CADA TESTE – SISTEMA *BOILER TURBINE*

Estudo	Mínima Função Objetivo	Máxima Função Objetivo	Desvio Padrão
I	3,23E+06	3,58E+06	9,51E+05
II	1,05E+07	1,97E+07	2,46E+06
III	8,68E+06	1,43E+07	1,51E+06
IV	8,92E+06	1,13E+07	7,41E+05

FONTE: DO AUTOR (2016)

A FIGURA 3.2 mostra graficamente as saídas dos estudos I, II, III e IV. Nota-se o Estudo II teve menos interferências e o que mais se aproximou das saídas de referências.

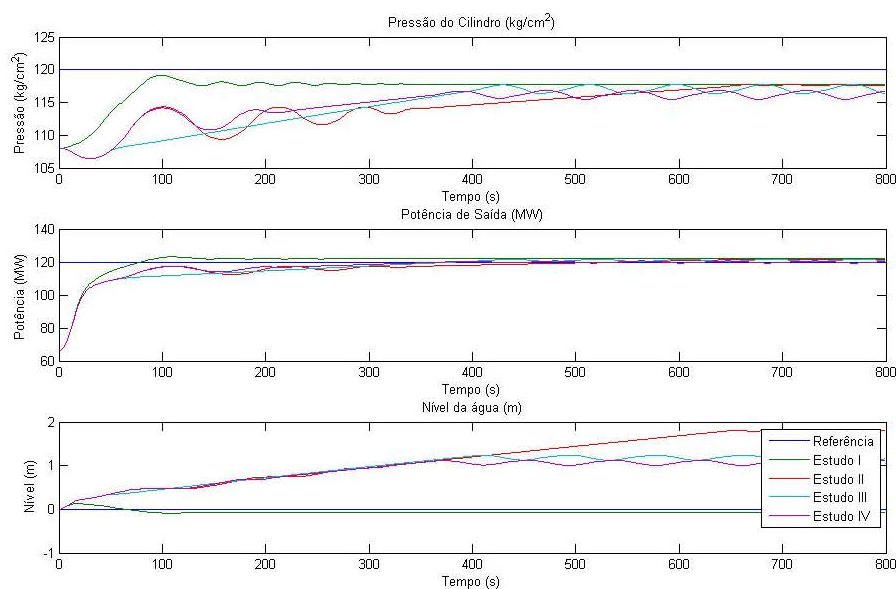


FIGURA 3.2 - COMPARATIVO DAS SAÍDAS DO *BOILER-TURBINE* DOS ESTUDOS I, II, III, E IV PARA DE CLÁSSICO

FONTE: DO AUTOR

A TABELA 3.4 mostra os valores de Wilcoxon-Mann-Whitney U-test dos testes realizados. Analisando a hipótese, nota-se que todos os estudos são parecidos (possuem respostas parecidas) e analisando o *p-values* o melhor valor encontrado é do Estudo I, coincidindo com a resposta gráfica.

TABELA 3.4 - COMPARATIVO POR WILCOXON-MANN-WHITNEY U-TEST ENTRE OS TESTES – SISTEMA *BOILER TURBINE*

		I	II	III	IV
Estudo I versus	<i>p-value</i>	-	7,06E-18	8,00E-17	3,19E-15
	h	-	1	1	1
Estudo II versus	<i>p-value</i>	7,06E-18	-	2,05E-07	5,64E-16
	h	1	-	1	1
Estudo III versus	<i>p-value</i>	8,00E-17	2,05E-07	-	7,46E-09
	h	1	1	-	1
Estudo IV versus	<i>p-value</i>	3,19E-15	5,64E-16	7,45E-09	-
	h	1	1	1	-

FONTE: DO AUTOR (2016)

A FIGURA 3.3 mostra o gráfico da mínima, máxima e mediana das melhores funções objetivo adquiridas, comprovando o indicado pelo teste de Wilcoxon-Mann-Whitney U-test e a resposta gráfica.

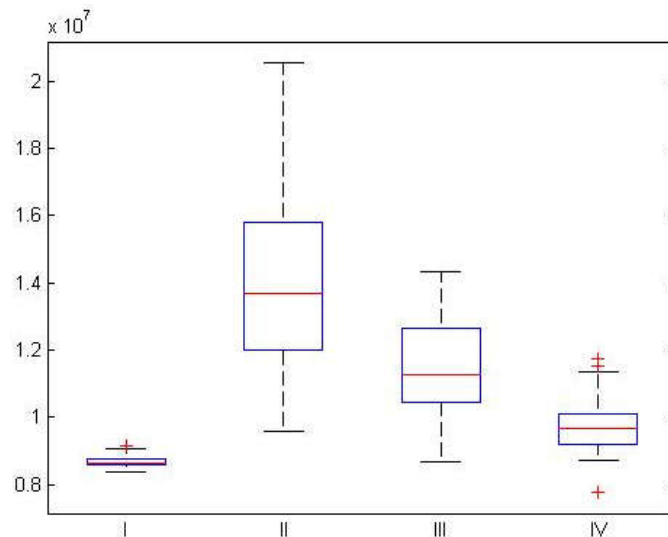


FIGURA 3.3 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS 4 TESTES - SISTEMA *BOILER TURBINE*

FONTE: DOA AUTOR (2016)

3.5.2 Resultados de Algoritmos Variantes do DE e outros algoritmos

A TABELA 3.5 mostra resultados de mínimo, máximo, média e desvio padrão obtidos, além de destacar os melhores resultados em cada coluna. Nesta tabela é possível analisar que entre todos os algoritmos o EPSDE teve o melhor desvio padrão de $7,42E+04$ e entre os algoritmos baseados em população o que apresentou o melhor desvio padrão foi o PSO com $3,52E+06$. Tanto o EPSDE, quanto o PSO apresentaram o mesmo valor mínimo da função objetivo de $3,20E+06$.

TABELA 3.5 - MÍNIMO, MÁXIMO, MÉDIA E DESVIO PADRÃO ENTRE ALGORITMOS - SISTEMA *BOILER TURBINE*

Algoritmo	Mínimo	Máximo	Média	Desvio Padrão
ALO	7,18E+06	5,33E+07	3,49E+07	1,52E+07
CMAES	1,05E+07	4,09E+09	2,87E+08	5,55E+08
CODE	3,88E+06	3,65E+07	7,29E+06	4,98E+06
DA	4,03E+06	5,82E+07	2,58E+07	1,36E+07
DE	3,23E+06	3,58E+06	3,37E+06	9,51E+04
EPSDE	3,20E+06	3,58E+06	3,22E+06	7,42E+04
GWO	3,89E+06	3,07E+07	1,03E+07	7,39E+06
JADE	3,89E+06	4,12E+07	1,17E+07	8,42E+06
jDE	5,09E+06	2,27E+07	1,14E+07	4,97E+06
MVO	6,31E+07	2,16E+09	1,08E+09	6,37E+08
PSO	3,20E+06	1,86E+07	4,65E+06	3,52E+06
SaDE	9,11E+06	4,00E+07	2,21E+07	5,16E+06

FONTE: DO AUTOR (2016)

A FIGURA 3.4 mostra a comparação das saídas entre os algoritmos DE clássico e suas ramificações para o estudo de caso *Boiler Turbine*.

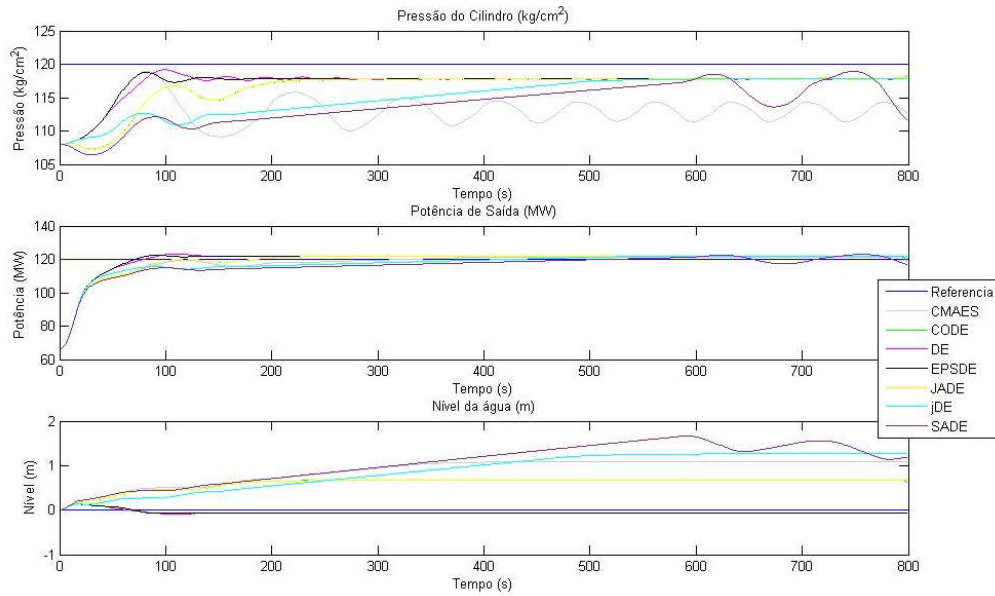


FIGURA 3.4 - COMPARATIVO ENTRE OS ALGORITMOS DE CLÁSSICO E SUAS RAMIFICAÇÕES – SISTEMA *BOILER TURBINE*
 FONTE: DO AUTOR

A FIGURA 3.5 mostra o comparativo das saídas entre o algoritmo clássico DE e outros algoritmos baseados em população para o *Boiler Turbine*.

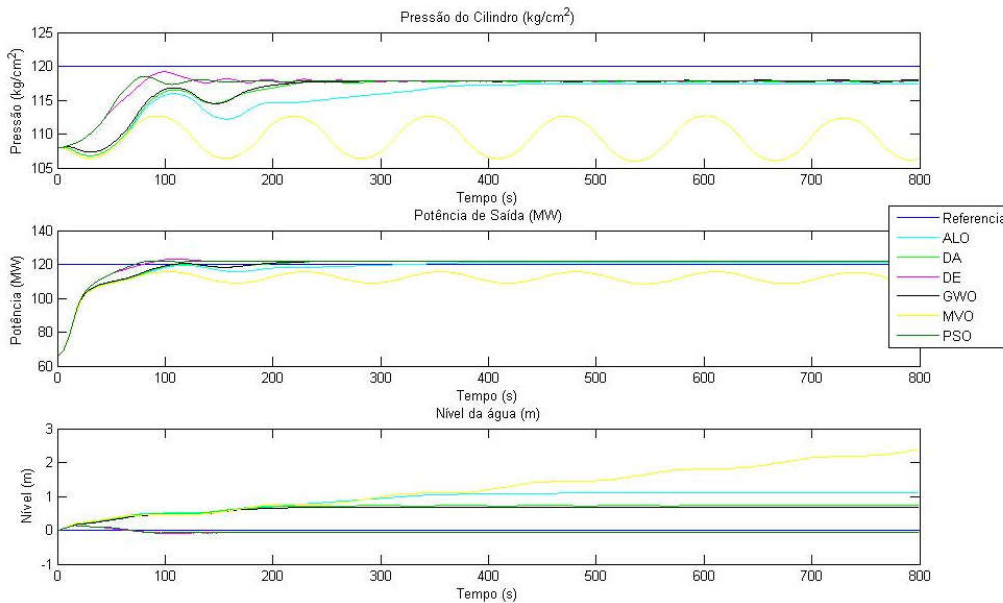


FIGURA 3.5 - COMPARATIVO ENTRE OS ALGORITMOS DE CLÁSSICO E ALGORITMOS BASEADOS EM POPULAÇÃO - SISTEMA *BOILER TURBINE*
 FONTE: DO AUTOR (2016)

Como os vetores são muito próximos, o teste de Wilcoxon-Mann-Whitney U-test não pode analisar os dados, assim é necessário outro teste de

hipótese (Teste T). A TABELA 3.6 mostra o comparativo do Teste T entre o algoritmo clássico DE e algoritmos ramificados do DE descritos nesta dissertação. Nela pode-se analisar que todos os algoritmos são parecidos entre si e que o SaDE é o algoritmo que mais rejeitou a hipótese nula.

TABELA 3.6 - TESTE T – DE CLÁSSICO VERSUS ALGORITMOS DE RAMIFICADOS – SISTEMA *BOILER TURBINE*

TABELA 3.7-TESTE		
DE versus algoritmos ramificados		
	<i>p-value</i>	h
CMAES	5,09E-11	1
CODE	1,30E-04	1
EPSDE	2,74E-07	1
JADE	2,57E-04	1
jDE	5,36E-05	1
SaDE	2,67E-13	1

FONTE: DO AUTOR (2016)

A FIGURA 3.6 mostra o gráfico da mínima, máxima e mediana das melhores funções comprovando o indicado pelo teste T e a resposta gráfica que o pior desempenho foi o algoritmo CMAES.

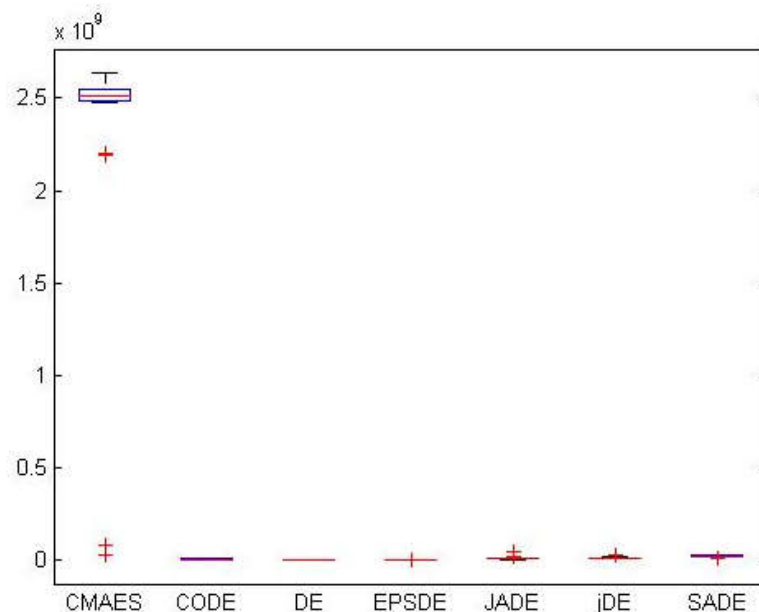


FIGURA 3.6 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS ALGORITMOS DERIVADOS DO DE - SISTEMA *BOILER TURBINE*
FONTE: DO AUTOR (2016)

A FIGURA 3.7 representa a FIGURA 3.6 sem o algoritmo CMAES para analisar qual teve o melhor desempenho. Nela pode-se verificar que o algoritmo com melhor desempenho foi o EPSDE.

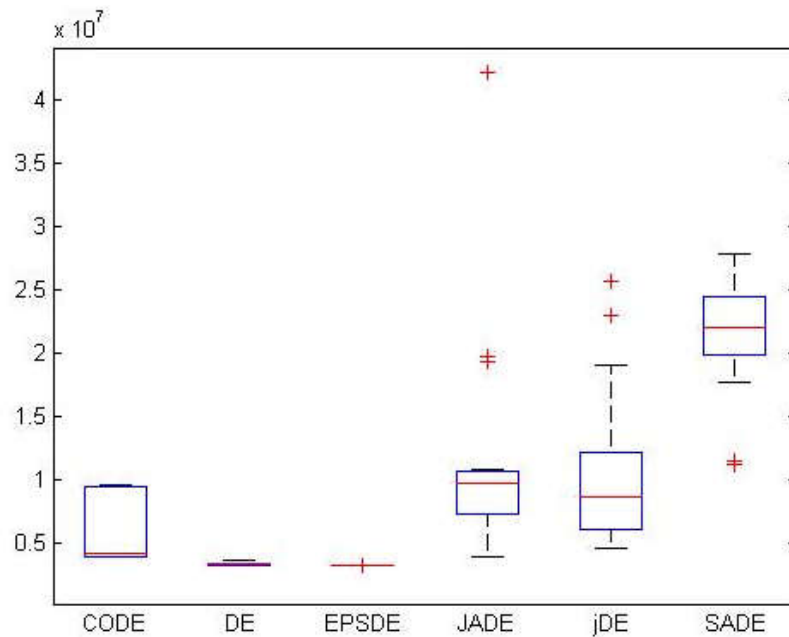


FIGURA 3.7 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS ALGORITMOS DERIVADOS DO DE SEM O CMAES - SISTEMA BOILER TURBINE
 FONTE: DO AUTOR (2016)

A TABELA 3.8 mostra o comparativo de Teste T entre o algoritmo clássico DE e algoritmos baseados em população descritos nesta dissertação. Nota-se que o melhor resultado foi o PSO com um p -value e $Zval$ aproximando-se de 1 (0,11 e -1,63, respectivamente), confirmando com a resposta gráfica e com o menor desvio padrão. O algoritmo que ficou mais longe do unitário foi o DA com p -value de 9,53E-07 e $Zval$ de -7,09.

TABELA 3.8 - TESTE T – DE CLÁSSICO VERSUS OUTROS ALGORITMOS BASEADOS EM POPULAÇÃO - SISTEMA *BOILER TURBINE*

DE versus algoritmos baseados em população			
	<i>p-value</i>	H	Zval
ALO	4,47E-06	1	-6,33
DA	9,53E-07	1	-7,09
GWO	2,27E-04	1	-4,53
MVO	1,77E-05	1	-5,68
PSO	0,11	1	-1,63

FONTE: DO AUTOR (2016)

A tabela de comparativo entre todos os algoritmos encontra-se no APÊNDICE 1.

A FIGURA 3.8 mostra o gráfico da mínima, máxima e mediana das melhores funções objetivo. Pode-se verificar que o melhor desempenho foi o algoritmo PSO, comparando-se ao clássico DE e o pior desempenho foi o MVO. A figura de comparação entre o algoritmo clássico DE, o EPSDE (melhor desempenho entre os derivados do DE) e o PSO pode ser vista no APÊNDICE 2.

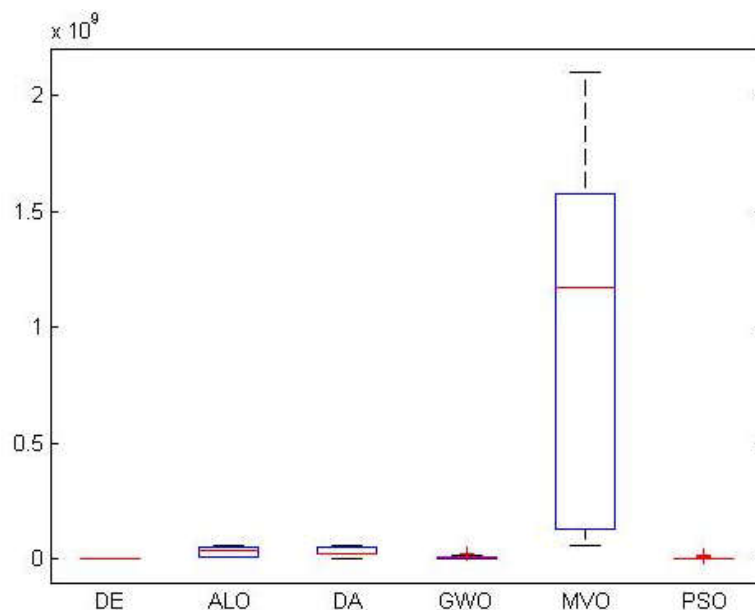


FIGURA 3.8 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DO ALGORITMO DE COMPARADO AOS ALGORITMOS BASEADOS EM POPULAÇÃO – SISTEMA *BOILER TURBINE*

A Figura 3.9 representa a FIGURA 3.8 sem o algoritmo MVO para analisar qual teve o melhor desempenho. Nela pode-se verificar que o algoritmo com melhor desempenho foi o PSO.

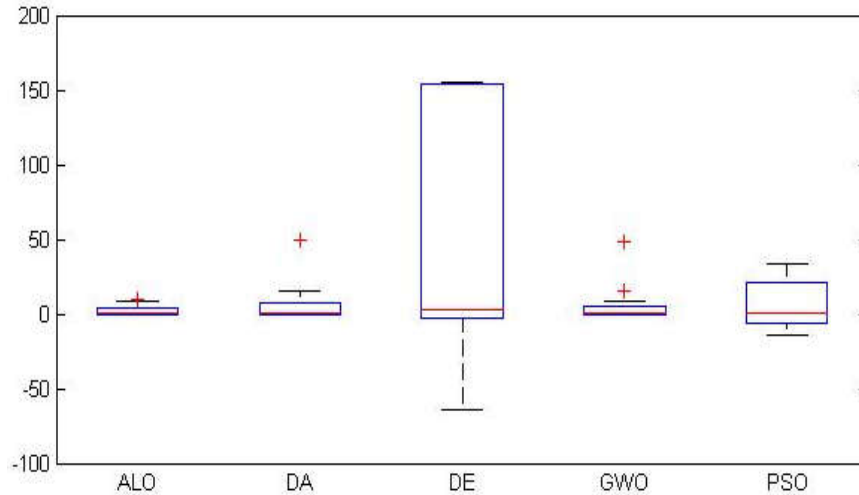


FIGURA 3.9 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DO ALGORITMO DE COMPARADO AOS ALGORITMOS BASEADOS EM POPULAÇÃO SEM MVO – SISTEMA BOILER TURBINE

3.6 ESTUDO DE CASO 2 – QUADRUPLE TANK

No segundo estudo de caso apresentado nesta dissertação apresenta um processo de laboratório originalmente apresentado por Johansson, 1997. O processo multivariável consistem em 4 tanques de água interconectados e duas válvulas para controle, como mostra a FIGURA 3.10. Seu objetivo é controlar o nível de dois tanques. As entradas do sistema são as tensões nas duas válvulas ($u = [v_1 \ v_2]^T$), as saídas do sistema são os níveis de água nos tanques inferiores em cm ($y = [y_1 \ y_2]^T$) e o vetor da variável de estado é o nível de todos os tanques ($x = [x_1 \ x_2 \ x_3 \ x_4]^T$), onde cada tanque possui um nível máximo de 20 cm.

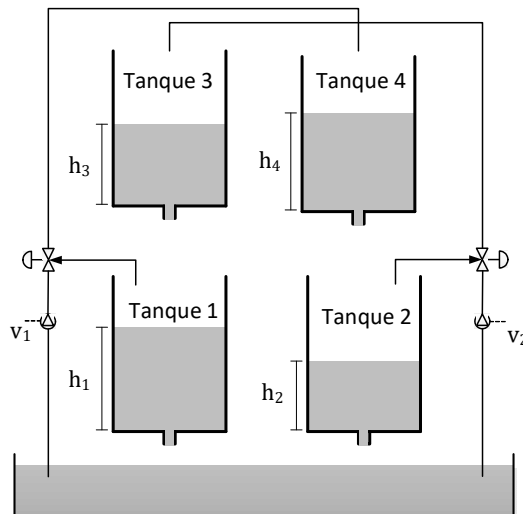


FIGURA 3.10 - ESQUEMA DO QUADRUPLE TANK
 FONTE: DO AUTOR

Esse sistema dinâmico é modelado pelas equações (3.16), (3.17), (3.18) e (3.19), tal que

$$\frac{d h_1}{dt} = \frac{a_1}{A_1} \sqrt{2g h_1} + \frac{a_3}{A_1} \sqrt{2g h_3} + \frac{\gamma_1 k_1}{A_1} v_1 \quad (3.16)$$

$$\frac{d h_2}{dt} = \frac{a_2}{A_2} \sqrt{2g h_2} + \frac{a_4}{A_2} \sqrt{2g h_4} + \frac{\gamma_2 k_2}{A_2} v_2 \quad (3.17)$$

$$\frac{d h_3}{dt} = \frac{a_3}{A_3} \sqrt{2g h_3} + \frac{(1 - \gamma_2) k_2}{A_3} v_2 \quad (3.18)$$

$$\frac{d h_4}{dt} = \frac{a_4}{A_4} \sqrt{2g h_4} + \frac{(1 - \gamma_1) k_1}{A_4} v_1 \quad (3.19)$$

onde A_i é a área de seção dos tanques (cm^2), a_i é a área de seção dos furos dos tanques (cm^2), h_i é o nível dos tanques (cm), u_i é a tensão nas válvulas de controle de vazão, ou seja, é o sinal de controle do sistema, $k_i u_i$ é a vazão, $\gamma_i \in (0,1)$ são os parâmetros que determina qual válvula é prioritária no experimento e g é a aceleração da gravidade (cm/s^2).

A aproximação linear do modelo é dada pela com a equação (3.20), tal que:

$$\begin{aligned} \frac{d}{dt} x &= A x + B u \\ y &= C x + D u \end{aligned} \quad (3.20)$$

onde A , B , C e D são dados por:

$$\begin{aligned}
 A &= \begin{bmatrix} \frac{\delta f_1}{\delta_1} & \frac{\delta f_1}{\delta_2} & \frac{\delta f_1}{\delta_3} & \frac{\delta f_1}{\delta_4} \\ \frac{\delta f_2}{\delta_1} & \frac{\delta f_2}{\delta_2} & \frac{\delta f_2}{\delta_3} & \frac{\delta f_2}{\delta_4} \\ \frac{\delta f_3}{\delta_1} & \frac{\delta f_3}{\delta_2} & \frac{\delta f_3}{\delta_3} & \frac{\delta f_3}{\delta_4} \\ \frac{\delta f_4}{\delta_1} & \frac{\delta f_4}{\delta_2} & \frac{\delta f_4}{\delta_3} & \frac{\delta f_4}{\delta_4} \end{bmatrix}, B = \begin{bmatrix} \frac{\delta f_1}{\delta u_1} & \frac{\delta f_1}{\delta u_2} \\ \frac{\delta f_2}{\delta u_1} & \frac{\delta f_2}{\delta u_2} \\ \frac{\delta f_3}{\delta u_1} & \frac{\delta f_3}{\delta u_2} \\ \frac{\delta f_4}{\delta u_1} & \frac{\delta f_4}{\delta u_2} \end{bmatrix}, \\
 C &= \begin{bmatrix} \frac{\delta y_1}{\delta_1} & \frac{\delta y_1}{\delta_2} & \frac{\delta y_1}{\delta_3} & \frac{\delta y_1}{\delta_4} \\ \frac{\delta y_2}{\delta_1} & \frac{\delta y_2}{\delta_2} & \frac{\delta y_2}{\delta_3} & \frac{\delta y_2}{\delta_4} \end{bmatrix} e D = \begin{bmatrix} \frac{\delta y_1}{\delta u_1} & \frac{\delta y_1}{\delta u_2} \\ \frac{\delta y_2}{\delta u_1} & \frac{\delta y_2}{\delta u_2} \end{bmatrix}
 \end{aligned}$$

Assim, a linearização do modelo em espaço de estados da equação em torno do ponto de equilíbrio é dada por:

$$\begin{aligned}
 x = \begin{bmatrix} \frac{1}{T_1} & 0 & \frac{A_4}{A_1 T_3} & 0 \\ 0 & \frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & \frac{1}{T_3} & \frac{1}{T_4} \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ \frac{(1 - \gamma_1) k_1}{A_4} & \frac{(1 - \gamma_2) k_2}{A_3} \\ 0 & 0 \end{bmatrix} u \quad (3.21)
 \end{aligned}$$

$$y = \begin{bmatrix} k_c & 0 & 0 & 0 \\ 0 & k_c & 0 & 0 \end{bmatrix} x$$

onde $x_i = i$, $u_i = v_i$, $y_i = y_i$, T_i é a constante de tempo dada por $T_i = \frac{A_i}{a_i} \sqrt{\frac{2h_i^0}{g}}$ e k_c é o ganho de cada válvula.

Os valores das características dos tanques e do ambiente são apresentados na TABELA 3.9.

TABELA 3.9 - VALORES DAS CARACTERÍSTICAS DOS TANQUES E AMBIENTE

Características	Valores
$[A_1, A_2, A_3, A_4]$	$[28; 32; 28; 32] \text{ cm}^2$
$[a_1, a_2, a_3, a_4]$	$[0,071; 0,057; 0,071; 0,057] \text{ cm}^2$
$[k_1, k_2]$	$[3,14; 3,29]$
k_c	0.5 V/cm
g	981 cm/s ²
$[\gamma_1, \gamma_2]$	$[0,35; 0,35]$

FONTE: KIRUBAKARAN (2014)

Os valores das condições iniciais utilizados nesta dissertação são mostrados na TABELA 3.10. As variáveis controláveis podem variar de 0 a 10 V, o qual corresponde ao valor do nível (0 a 20 cm).

TABELA 3.10 - VALORES DE CONDIÇÕES INICIAIS DO SISTEMA DE QUADRUPLE TANK

Parâmetro	Valor
x_1^0	12,6
x_2^0	13
x_3^0	4,8
x_4^0	4,9
u_1^0	3,15
u_2^0	3,15

FONTE: JOHANSSON (2000)

Assim como em *boiler turbine*, há muitas técnicas utilizadas atualmente para controlar o sistema de *quadruple tank*, tais como mpMPC (do inglês, *multi-parametric model predictive controller*), utilizado por Kirubakaran (KURIBAKARAN et al., 2014) no controle PI. No estudo de Kirubakaran, é comparado o desempenho de 5 derivações do mpMPC: C-mpMPC (*centralized mpMPC*), DC-mpMPC (*decentralized mpMPC*), CD-mpMPC (*cooperative distributed mpMPC*), DC-PI (controle proporcional integral desacoplado), CG-mpMPC (*cooperative game mpMPC*), com o C-mpMPC sendo o mais rápido para estabilizar a resposta. Também há trabalhos que alteram algum parâmetro original do sistema como é o caso de Sutha (SUTHA et al., 2015) que usa o MLS (do inglês, *multi-level switching*) para proteger o controle da válvula. Neste artigo, o modelo do *quadruple tank* é modificado inserindo uma válvulas no tanque 3 e outra válvula no tanque 4 controlada manualmente, causando assim, uma dinâmica adicional não-linear ao modelo original e o PI acoplado é otimizado pelos algoritmos.

3.6.1 Resultados usando algoritmo DE clássico

A TABELA 3.11 mostra a mínima e máxima função objetivo e o desvio padrão para os 4 conjuntos de testes realizados. O melhor desvio padrão e também a menor função objetivo é apresentado no estudo IV.

TABELA 3.11 - VALORES DE MÍNIMA E MÁXIMA FUNÇÃO OBJETIVO E DESVIO PADRÃO PARA CADA TESTE - SISTEMA *QUADRUPLE TANK*

Estudo	Mínima Função Objetivo	Máxima Função Objetivo	Desvio Padrão
I	6,10E+03	483,20E+03	124,24E+03
II	15,44E+03	424,51E+03	64,73E+03
III	5,35E+03	424,51E+03	106,61E+03
IV	5,00E+03	169,46E+03	31,98E+03

FONTE: DO AUTOR (2016)

A FIGURA 3.11 mostra o comparativo das saídas do sistema de *quadruple tank* para os testes I, II, III e IV. Nota-se graficamente que o Estudo I teve melhor desempenho, porém teve maiores erros quando comparado a outros estudos no gráfico de nível do tanque 1.

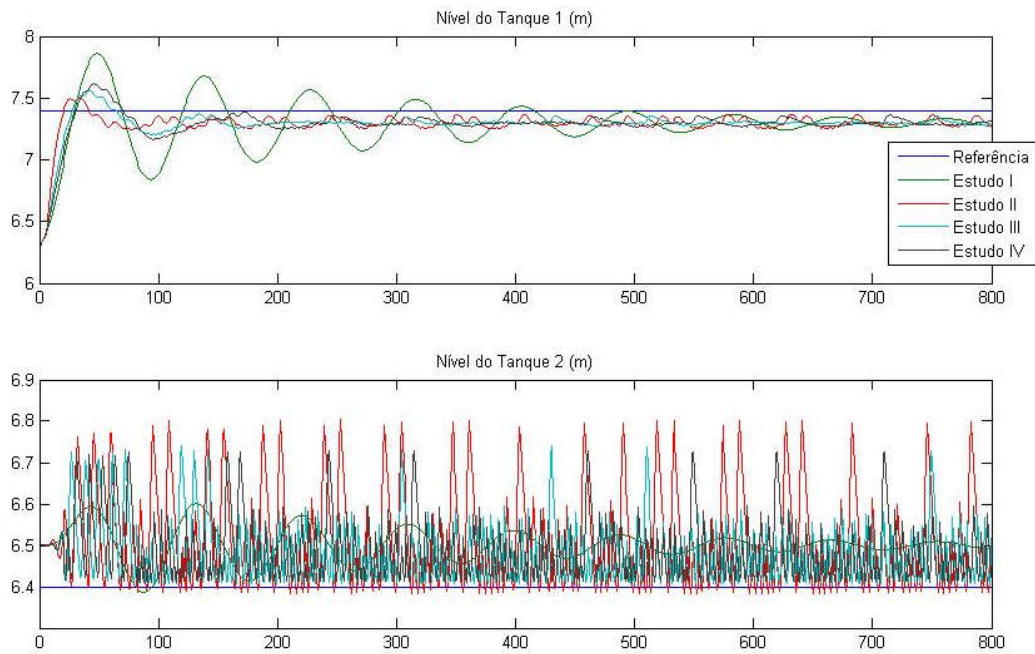


FIGURA 3.11 - COMPARATIVO DA SAÍDA DO *QUADRUPLE TANK* DOS TESTES I, II, III E IV PARA DE CLÁSSICO
 FONTE: DO AUTOR

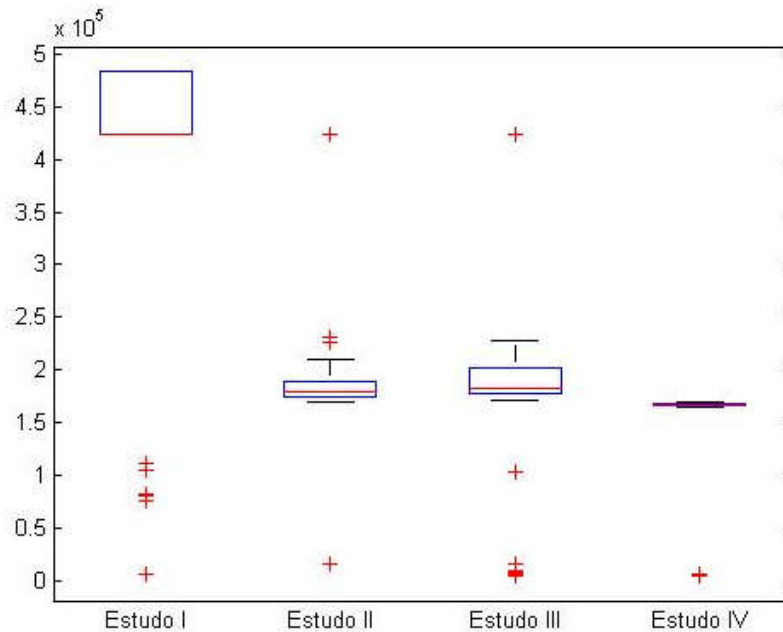
A TABELA 3.12 mostra o teste do Wilcoxon-Mann-Whitney U-test aplicado aos 4 testes realizados para o *quadruple tank*, podendo ser notado que o estudo II rejeitou a hipótese nula quando comparado ao estudo III, ou seja, que as duas respostas são completamente diferentes (pode ser visto pelo gráfico do nível do tanque 1 que a saída obtida no estudo II é o oposto do estudo III). Já observando o *p-value*, é possível comprovar que o estudo IV é o que obteve o melhor desempenho.

TABELA 3.12 - COMPARATIVO POR WILCOXON-MANN-WHITNEY U-TEST ENTRE OS TESTES – SISTEMA *QUADRUPLE TANK*

		I	II	III	IV
Estudo I	<i>p-value</i>	-	1,58E-09	4,15E-08	3,39E-11
versus	h	-	1	1	1
Estudo II	<i>p-value</i>	1,58E-09	-	0,16	1,51E-16
versus	h	1	-	0	1
Estudo III	<i>p-value</i>	4,15E-08	0,16	-	3,55E-11
versus	H	1	0	-	1
Estudo IV	<i>p-value</i>	3,39E-11	1,51E-16	3,55E-11	-
versus	h	1	1	1	-

FONTE: DO AUTOR (2016)

A FIGURA 3.13 mostra o gráfico da mínima, máxima e mediana das melhores funções objetivo comprovando os resultados obtidos pelo teste de Wilcoxon e a resposta gráfica.

FIGURA 3.12 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS 4 ESTUDOS - SISTEMA *BOILER TURBINE*

FONTE: DO AUTOR (2016)

3.6.2 Resultados de Algoritmos Variantes do DE e outros algoritmos

A TABELA 3.13 mostra resultados de mínimo, máximo, média e desvio padrão obtidos para o *quadruple tank*. Nesta tabela é possível analisar que entre todos os algoritmos o EPSDE teve o melhor desvio padrão de $5,52E+03$ e entre os algoritmos baseados em população o que apresentou o melhor desvio padrão foi o PSO com $9,00E+03$. O PSO apresentou o melhor valor quando comparado o mínimo da função objetivo de 28,74. Tanto o EPSDE quanto o PSO apresentaram o mesmo valor de máximo da função objetivo de $3,78E+04$. No entanto o EPSDE apresentou a melhor média de $2,09E+04$.

TABELA 3.13 - MÍNIMO, MÁXIMO, MÉDIA E DESVIO PADRÃO ENTRE ALGORITMOS - SISTEMA QUADRUPLE TANK

Algoritmo	Mínimo	Máximo	Média	Desvio Padrão
ALO	75,47	4,83E+05	4,71E+05	6,89E+04
CMAES	17,20	7,60E+06	5,85E+05	1,48E+06
CODE	48,62	4,83E+05	1,74E+05	2,06E+05
DA	4,21E+05	4,83E+05	4,59E+05	3,01E+04
DE	6,10E+03	4,83E+05	4,00E+05	1,24E+05
EPSDE	864,78	3,78E+04	2,09E+04	5,52E+03
GWO	4,21E+05	4,83E+05	4,79E+05	1,48E+04
JADE	42,12	4,83E+05	2,51E+05	2,21E+05
jDE	38,41	4,83E+05	3,31E+05	1,63E+05
MVO	6,99E+04	7,59E+05	4,81E+05	7,62E+04
PSO	28,74	3,78E+04	2,36E+04	9,00E+03
SaDE	27,74	4,86E+05	1,28E+05	1,75E+05

FONTE: DO AUTOR (2016)

A FIGURA 3.13 mostra a comparação das saídas entre os algoritmos DE clássico mais estável e suas ramificações para o estudo de caso *quadruple tank*.

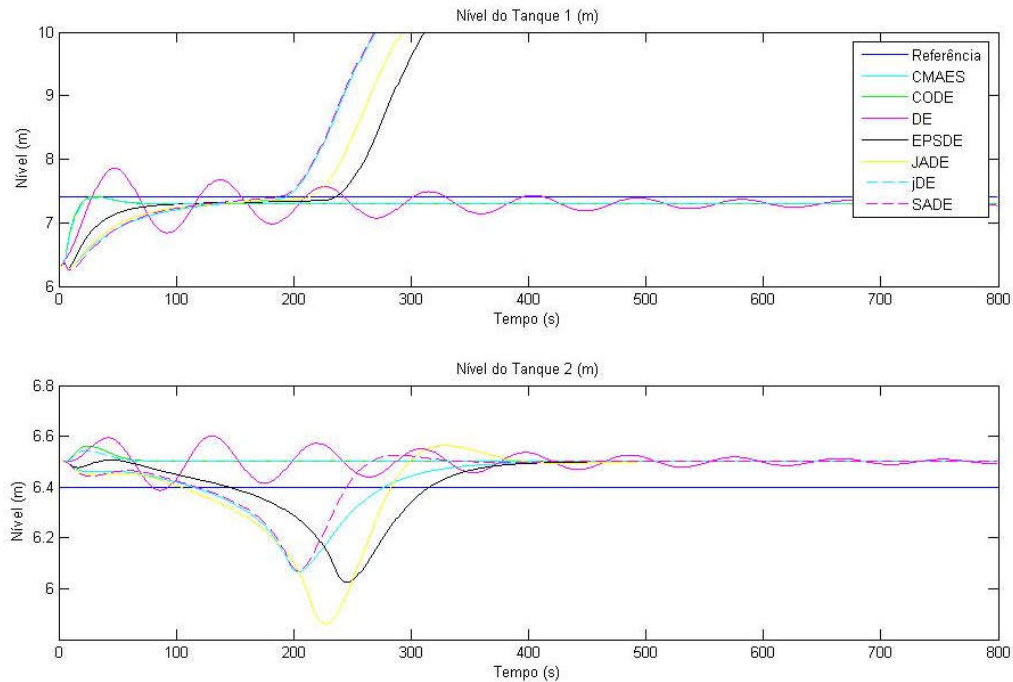


FIGURA 3.13 - COMPARATIVO ENTRE OS ALGORITMOS DE CLÁSSICO E SUAS RAMIFICAÇÕES – SISTEMA QUADRUPLE TANK
 FONTE: DO AUTOR

A FIGURA 3.14 mostra o comparativo das saídas entre o algoritmo clássico DE e outros algoritmos baseados em população para o *quadruple tank*.

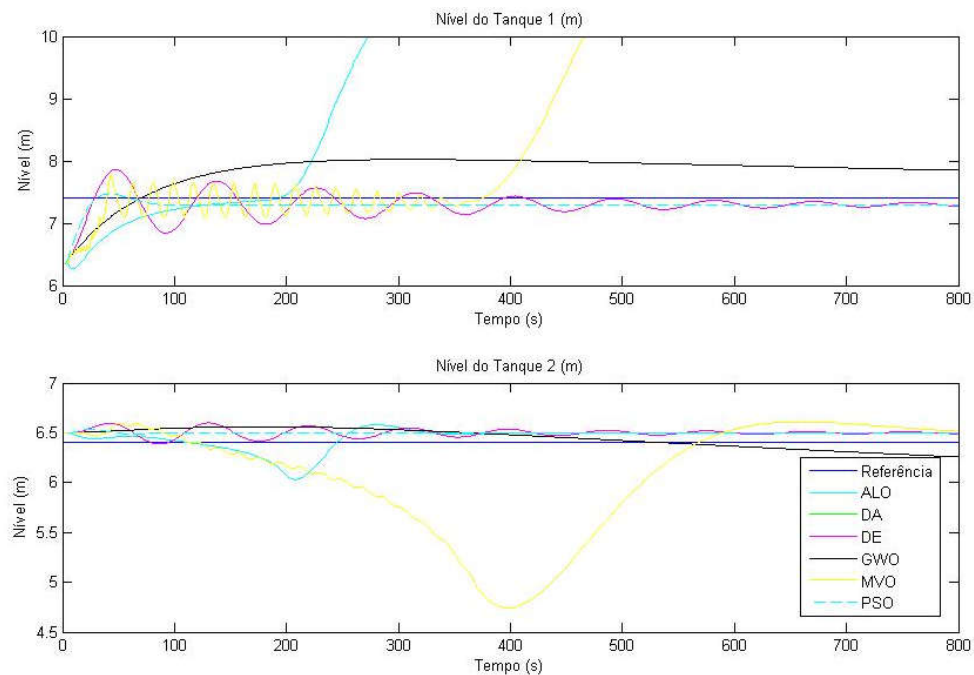


FIGURA 3.14 - COMPARATIVO ENTRE OS ALGORITMOS DE CLÁSSICO E ALGORITMOS BASEADOS EM POPULAÇÃO - SISTEMA QUADRUPLE TANK
 FONTE: DO AUTOR (2016)

A TABELA 3.14 mostra o comparativo de Wilcoxon-Mann-Whitney U-test entre o algoritmo clássico DE e algoritmos ramificados do DE descritos nesta dissertação. Observa-se nesta tabela que o CMAES e o JADE rejeitaram a hipótese nula quando comparado ao clássico DE e também apresentaram o maior desvio padrão.

TABELA 3.14 - TESTE DE WILCOXON-MANN-WHITNEY U-TEST – DE CLÁSSICO VERSUS ALGORITMOS DE RAMIFICADOS – SISTEMA *QUADRUPLE TANK*

DE versus algoritmos ramificados		
	<i>p-value</i>	h
CMAES	0,10	0
CODE	6,12E-10	1
EPSDE	1,27E-16	1
JADE	0,15	0
jDE	0,006	1
SaDE	8,95E-11	1

FONTE: DO AUTOR (2016)

A FIGURA 3.15 mostra o gráfico da mínima, máxima e mediana das melhores funções objetivo adquiridas comprovando o indicado pelo teste de Wilcoxon-Mann-Whitney U-test e a resposta gráfica que todos os algoritmos derivados do DE clássico tem máxima e mínima próximos, porém o pior desempenho foi o algoritmo CMAES.

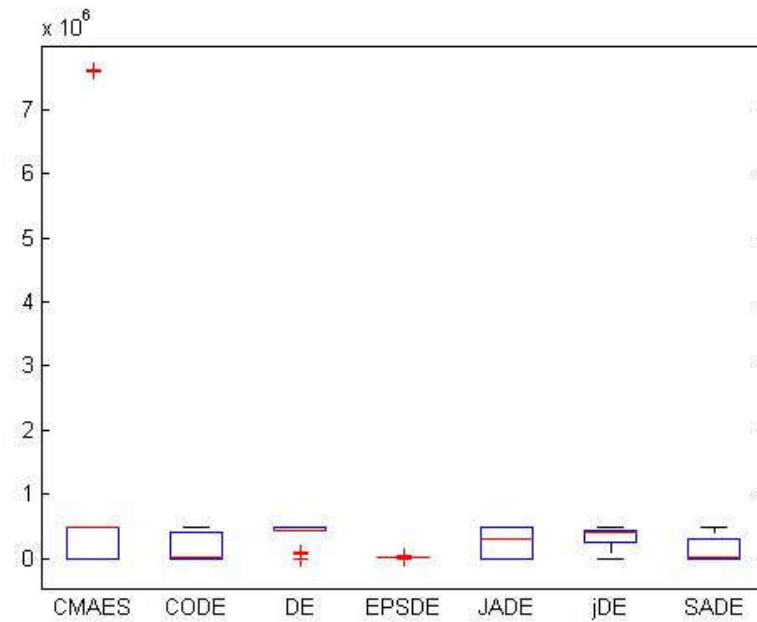


FIGURA 3.15 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS ALGORITMOS DERIVADOS DO DE - SISTEMA *QUADRUPLE TANK*
 FONTE: DO AUTOR (2016)

A FIGURA 3.16 representa a FIGURA 3.15 sem o algoritmo CMAES para analisar qual teve o melhor desempenho. Nela pode-se verificar que o algoritmo com melhor desempenho foi o EPSDE, pois teve o menor desvio padrão e também a menor mínima.

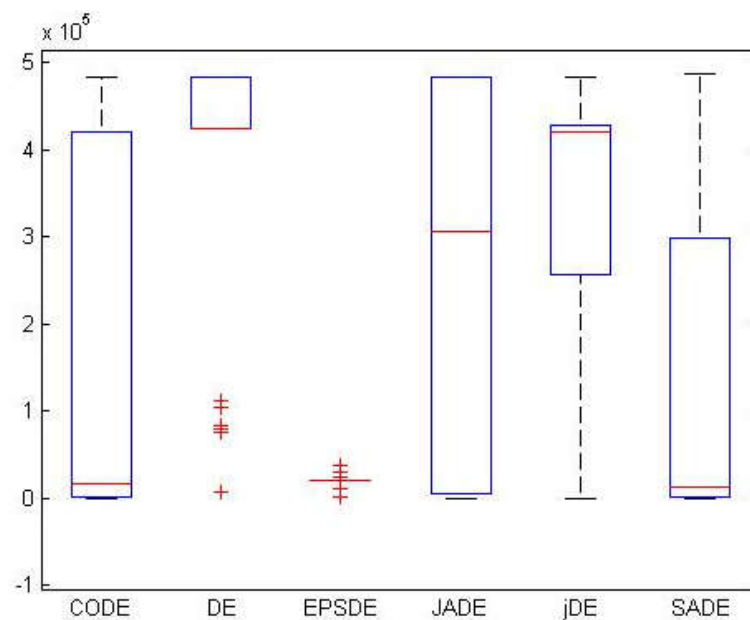


FIGURA 3.16 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DOS ALGORITMOS DERIVADOS DO DE SEM O CMAES - SISTEMA *QUADRUPLE TANK*
 FONTE: DO AUTOR (2016)

A TABELA 3.15 mostra o comparativo de Wilcoxon-Mann-Whitney U-test entre o algoritmo clássico DE e algoritmos baseados em população descritos nesta dissertação. Nota-se pelo *p-value* que o melhor resultado foi o PSO, confirmando com a resposta gráfica e com o menor desvio padrão.

TABELA 3.15 - TESTE DE WILCOXON-MANN-WHITNEY U-TEST – DE CLÁSSICO VERSUS OUTROS ALGORITMOS BASEADOS EM POPULAÇÃO - SISTEMA *QUADRUPLE TANK*
DE versus algoritmos baseados em população

	p-value	H
ALO	1,14E-11	1
DA	0,009	1
GWO	0,004	1
MVO	3,98E-15	1
PSO	1,20E-16	1

FONTE: DO AUTOR (2016)

A tabela de comparativo entre todos os algoritmos encontra-se no APÊNDICE 1.

A FIGURA 3.17 mostra o gráfico da mínima, máxima e mediana das melhores funções objetivo adquiridas realizando-se os 20 experimentos de cada um dos algoritmos baseados em população e o algoritmo clássico DE. Pode-se verificar que o melhor desempenho foi o algoritmo PSO, comparando-se ao clássico DE e o pior desempenho foi o MVO. A figura de comparação entre o algoritmo clássico DE, o EPSDE (melhor desempenho entre os derivados do DE) e o PSO pode ser vista no APÊNDICE 2.

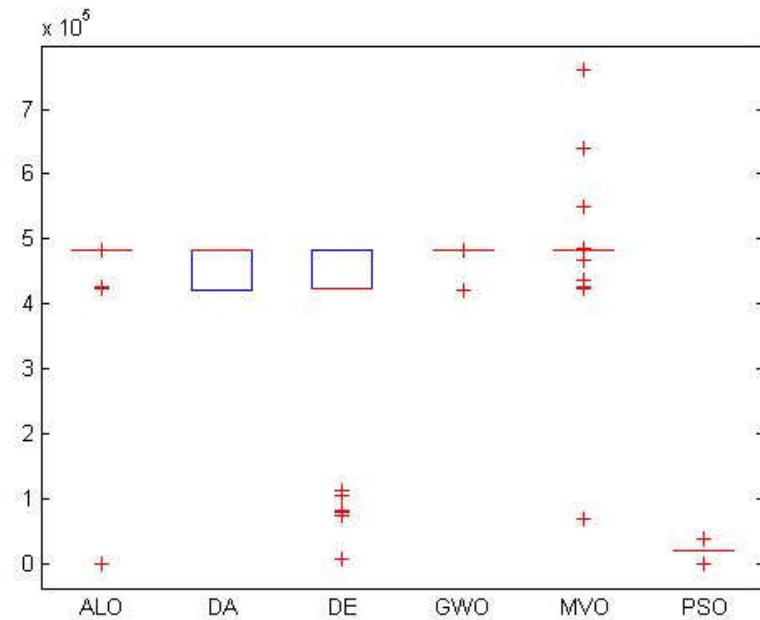


FIGURA 3.17 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DO ALGORITMO DE COMPARADO AOS ALGORITMOS BASEADOS EM POPULAÇÃO – SISTEMA QUADRUPLE TANK

FONTE: DO AUTOR (2016)

A FIGURA 3.18 representa a FIGURA 3.17 sem o algoritmo MVO para analisar qual teve o melhor desempenho. Nela pode-se verificar que o algoritmo com melhor desempenho foi o DE clássico, pois teve o menor desvio padrão e também a menor mínima.

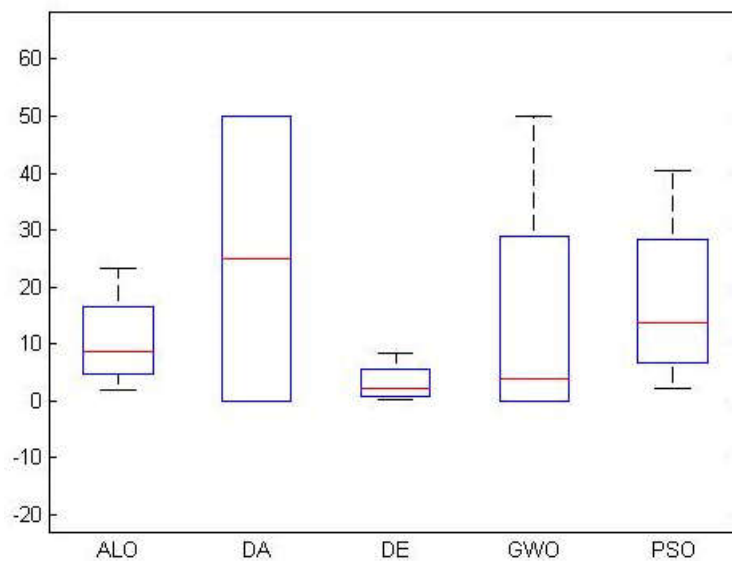


FIGURA 3.18 - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA DO ALGORITMO DE COMPARADO AOS ALGORITMOS BASEADOS EM POPULAÇÃO SEM MVO – SISTEMA QUADRUPLE TANK

FONTE: DO AUTOR (2016)

4 CONCLUSÃO

Como os algoritmos DE têm apresentado bons resultados quando aplicados a sistemas complexos, cada vez mais variantes do DE clássico tem sido desenvolvidas. Nesta dissertação foram testadas algumas dessas variantes aos estudos de caso *Boiler Turbine* e *Quadruple Tank* e comparados a outros algoritmos baseados em população.

Os estudos de casos mencionados são sistemas multiobjetivos que foram tratados como um sistema *a-priori*, ou seja, como um mono-objetivo. Para isso, foi utilizado valores e plantas originais e suas funções objetivos foram somadas e ponderadas.

Os algoritmos mencionados foram utilizados para otimizar aos parâmetros dos controladores PID acoplados que controlam as plantas dos estudos de caso. Como esperado, os algoritmos DE (clássico e variantes) apresentaram um bom desempenho nos dois estudos de casos mencionados, comprovando sua eficácia a sistemas multiobjetivos ponderados.

Para a otimização utilizando o DE, foi realizado 4 testes alterando-se os parâmetros de fator de cruzamento (Cr) e mutação (F). Para o *Boiler Turbine* o melhor valor de desvio padrão ($7,41E+05$ contra o pior valor de desvio padrão de $2,46E+06$) foi encontrado quando o fator de cruzamento foi maior que a mutação (0,8; 0,6, respectivamente), já para o *Quadruple Tank*, o melhor valor de desvio padrão ($3,19E+03$ contra o pior valor de desvio padrão de $124,24E+03$) foi encontrado quando a mutação e o fator de cruzamento foram iguais (0,8; 0,8, respectivamente).

Em comparação a outros algoritmos, tanto para o *Boiler Turbine* quanto para o *Quadruple Tank*, o algoritmo de EPSDE teve o melhor resultado quando comparado o desvio padrão ($1,89E+03$ contra $7,63E+08$ e $5,52E+03$ contra $2,21E+04$, respectivamente). O algoritmo que apresentou o pior desempenho em relação ao desvio padrão tanto para o *Boiler Turbine* quanto para o *Quadruple Tank* foi o CMAES.

Para trabalhos futuros, o objetivo é aplicar metaheurísticas de otimização aos estudos de casos tratando-os como sistemas multiobjectivos e também testar outras metaheurísticas variantes do DE.

REFERÊNCIAS

ABRAHAM, A.; JAIN, L; GOLDBERG R. **Evolutionary Multiobjective Optimization: Theoretical Advances and Applications**. London, UK: Springer, 2005.

ACHARJEE, P. **Optimal Power Flow with UPFC Using Security Constrained Self-Adaptive Differential Evolutionary Algorithm for Restructures Power System**. International Journal of Electrical Power & Energy Systems, v. 76, p. 69-81, 2016.

ANDERSON, M.; GUNNARSSON, S.; GLAD, T.; NORRLÖF, M. **A Simulation and Animation Tool for Studying Multivariable Control**. World Congress, Linköping, Suécia, v. 15, n. 1, p.1432-1432, 2002.

ARAÚJO, R. A.; OLIVEIRA, A. L. I.; MEIRA, S. **A Learning Process based on Covariance Matrix Adaptation for Morphological-Linear Perceptrons**. IEEE Congress on Evolutionary Computation, Cancún, México, p. 2275-2282, 2013.

ASCIONE, F.; BIANCO, N.; STASIO, C. MAURO, G. M.; VANOLI, G.P. **Simulation-Based Model Predictive Control by Multi-Objective Optimization of Building Energy Performance and Thermal Comfort**. Energy and Buildings, v. 11, p. 131-144, 2016.

AZUMA, R.M. **Otimização Multiobjetivo em Problema de Estoque e Roteamento Gerenciados pelo Fornecedor**. 2011. 99 f. Dissertação (Mestre em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Departamento de Engenharia de Computação e Automação Industrial, Universidade Estadual de Campinas, Campinas, SP, 2011.

BELL, R.D; ÅSTRÖM, K.J. **Dynamic Models for Boiler-Turbine-Alternator Units: Data logs and parameter estimation for 160MW unit**. Lund Institute of Technology, Sweden, 1987.

BHATTACHARYA, S. K. **Control System Engineering**. 2.ed. India: Pearson Education, 2009.

BHASKAR, V.; GUPTA, S.K.; RAY, A.K. **Applications of Multiobjective Optimization in Chemical Engineering**. University of Singapore, Singapore, 2014.

BISWAS, P. P.; SRIVASTAVA, R.; RAY, S.; SAMANTA, A, N. **Sliding Mode Control of Quadruple Tank Process**. *Mechatronics*, v. 19, n. 4, p. 548-561, 2009.

BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. **A Survey on Optimization Metaheuristics**. *Information Sciences*, v. 237, p. 82-117, 2013.

BREST, J.; ZAMUDA, A.; BOSKOVIC, B.; GREINER, S.; ZUMER, V. **An Analysis of the Control Parameters' Adaptation in DE**. Institute of Computer Science, Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, Eslovênia, 2000.

BREST, J.; GREINER, S.; BOSKOVIC, B.; MERNIK, M.; ZUMER, V. **Self-adaptive control Parameters in Differential Evolution: a comparative study on numerical benchmark problem**. *IEEE Transactions on Evolutionary Computation*, v. 10, p. 646-657, 2006.

CAVA, W.; DANAI, K.; SPECTOR, L.; FLEMING, P.; WRIGHT, A.; LACKNER, M. **Automatic Identification of Wind Turbine Models Using Evolutionary Multiobjective Optimization**. *Renewable Energy*, v. 87, n. 2, p. 892-902, 2016.

CHAKRABORT, U.K. **Advances in Differential Evolution**. USA: Missouri, 2008.

COELLO C. A. C. **An Introduction to Multi-objective Particle Swarm Optimizers**. *Soft Computing in Industrial Applications*, v. 96, p. 3-12, 2011.

COELLO, C. A. C. LAMONT, G.B.; VELDHUIZEN, D.A. **Evolutionary Algorithms for Solving Multi-Objective Problems**. USA: Springer Science and Business Media, 2007.

COELHO, L. S. **Otimização por Enxame de Partículas (Particle Swarm Optimization, PSO)**. Parte de mini-curso apresentado na 1ª Escola Luso-Brasileira de Computação Evolutiva, Belo Horizonte, MG, 2009.

DAS, S.; SUGANTHAN, P. N. **Differential Evolution: A survey of the State-of-the-Art**. *IEEE Transactions on Evolutionary Computation*, v. 15, n. 1, p.4-31, 2011.

DAS, S.; ABRAHAM, A.; CHAKRABORTY, U. K.; KONAR, A. **Differential Evolution Using a Neighborhood-based Mutation Operator**. *IEEE Transactions on Evolutionary Computation*, v. 13, p. 526-553, 2009.

DEB, K. **Multi-Objective Optimization Using Evolutionary Algorithms**. England: John Wiley & Sons, Ltd, 2001.

DELBEM, A. C. B.; GABRIEL, P. H. R. **Fundamentos de Algoritmos Evolutivos**. Disponível em: <http://www.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_ND_75.pdf>. Acesso em: 05/01/2016.

DIMEO, R. **Boiler-Turbine Control System Design Using a Genetic Algorithm**. IEEE Transactions on Energy Conversion, v. 10, n. 4, p. 752-759, 1995.

FALCO, I.; SCAFURI, U.; TARANTINO, E. **Mapping of Time-Consuming Multitask Applications on a Cloud System by Multiobjective Differential Evolution**. Parallel Computing, v. 18, p. 40-50, 2015.

FATEEN, S. K.; PETRICIOLET, A., RANGAIAH, G. P.N. **Evaluation of Covariance Matrix Adaptation Evolution Strategy, Shuffled Complex Evolution and Firefly Algorithms for Phase Stability, Phase Equilibrium and Chemical Equilibrium Problems**. Chemical Engineering Research and Design, v. 90, p. 2051-2071, 2012.

FERMINO, F. **Estudo Comparativo de Métodos de Sintonia de Controladores PID**. 90 f. Trabalho de Graduação (Engenharia Elétrica com ênfase em Sistemas de Energia e Automação) – Setor de Engenharia Elétrica, Universidade de São Carlos, São Carlos, 2014.

GABRIEL, P.H.R.; DELBEM, A.C.B. **Fundamentos de Algoritmos Evolutivos**. 2014.

GARRIDO, J.; VÁZQUEZ, F.; MORILLA, F. **Centralized Multivariable Control by Simplified Decoupling**. Journal of Process Control, v. 22, p. 1044-1062, 2012.

GHABRAEI, S.; MORADI, H.; VOSSOUGH, G. **Multivariable Robust Adaptive Sliding Mode Control of an Industrial Boiler-Turbine in the Presence of Modeling in Precisions and External Disturbances: A comparison with Type-I Servo Controller**. ISA Transactions, v. 58, p. 398-408, 2015.

GHOSH, A.; CHATTOPADHYAY, S. **Trajectory Optimization of Satellite Launch Vehicle using Self Adaptive Differential Evolution Algorithm**. IEEE Power, Communication and Information Technology Conference (PCITC), Bhubaneswar, India, v.1, p. 297-301, 2015.

GOMES, J.H.F.; COSTA, S.C.; PAIVA, A.P.; BALESTRASSI, P.P. **Otimização de Múltiplos Objetivos na Soldagem de Revestimento de Chapas de Aço Carbono ABNT 1020 Utilizando Arame Tubular Inoxidável Austenítico**. Soldagem & Inspeção, v.16, n.6, p. 232-242, 2011.

HANSEN, N.; OSTERMEIER, A. **Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation**. IEEE International Conference on Evolutionary Computation, p. 312-317, Nagoya, Japão, 1996.

IACCA, G.; CARAFFINI, F., NERI, F. **Continuous Parameter Pools in Ensemble Self-Adaptive Differential Evolution**. IEEE Symposium Series on Computation Intelligence. v. 1, p. 1529-1536, Cidade do Cabo, Africa do Sul, 2015.

IRUTHAYARAJAN, M.W. **Evolutionary Algorithms Based Design of Multivariable PID Controller**. Expert System with Applications, v.36, n.5, p. 9159-9167, 2009.

IRUTHAYARAJAN, M.W; BASKAR, S. **Covariance Matrix Adaptation Evolution Strategy Based Design of Centralized PID Controller**. Expert System with Applications, v. 37, p. 5775-5781, 2010.

JOHANSSON, K.H. **The Quadruple-Tank Process: A Multivariable Laboratory Process with an Adjustable Zero**. IEEE Transactions on Control Systems Technology. v. 8, n. 3, p. 456-465, 2000.

KENNEDY, J.F.; EBERHART, R.C.; SHI, Y. **Particle Swarm Optimization**. IEEE int'l conf. on neural networks, v. 4, p. 1942-1948, 1995.

KIRUBAKARAN, V.; RADHAKRSHNAN, T. K.; SIVAKIMARAN, N. **Distributed Multiparametric Model Predictive Control Design for a Quadruple Tank Process**. Measurement, v. 47, p. 841-854, 2014.

KOCAARSLAN, I.; ÇAM, E. **An Adaptive Control Application in a Large Thermal Combined Power Plant**. Energy Conversion and Management, v. 48, n. 1, p. 174-183, 2007.

KONAK A.; COIT D. W.; SMITH A. E. **Multi-objective Optimization Using Genetic Algorithms: A Tutorial**. Reliability Engineering & System Safety, p. 922-1007, 2006.
KONG, X.; LIU, X. LEE, K. Y. **Nonlinear Multivariable Hierarchical Model Predictive Control for Boiler-Turbine System**. Energy, v. 93, n. 1, p. 309-322, 2015.

KRAUSE, J.; PARPINELLI, R.S.; LOPES, H.S. **Proposta de um Algoritmo Inspirado em Evolução Diferencial Aplicado ao Problema Multidimensional da Mochila**. XIX Encontro Nacional de Inteligência Artificial, Laboratório de Bioinformática, Universidade Tecnológica Federal do Paraná, Curitiba, 2012.

LI, K.; KWONG, S. **A General Framework for Evolutionary Multiobjective Optimization via Manifold Learning**. Neurocomputing, v. 146, p. 65-74, 2014.

LI, Z.; XU, X.; DENG, S.; PAN, D. **A Novel Proportional-Derivative (PD) Law Based Fuzzy Logic Principles Assisted Controller for Simultaneously Controlling Indoor Temperature and Humidity Using a Direct Expansion (DX) Air Conditioning (A/C) System**. International Journal of Refrigeration, v. 57, p. 239-256, 2015.

LIN, Q.; ZHU, Q.; HUANG, P.; CHEN, J.; MING, Z.; YU, J. **A Novel Hybrid Multi-Objective Immune Algorithm with Adaptive Differential Evolution**. Computer & Operations Research, v. 62, p. 95-111, 2015.

LIU, T.; GAO, X.; WANG, L. **Multi-Objective Optimization Method Using an Improved NSGA-II Algorithm for Oil-Gas Production Process**. Journal of the Taiwan Institute of Chemical Engineers, v. 57, p. 42-53, 2015.

LU, Z.; LIN, W. FENG, G.; WAN, F. **A Study of Nonlinear Control Schemes for a Boiler-Turbine Unit**. IFAC Proceedings Volumes, v. 43, n. 14, p. 1368-1373, 2010.

MAHDAVI, S.; SIRI, M. E.; RAHNAMEYAN, S. **Metaheuristics in Large-Scale Global Continues Optimization: A survey**. Information Sciences, v.295, p. 407-428, 2015.

MALLIPEDDI, R; SUGANTHAN, P; PAN, Q; TASGETIREN, M. **Differential Evolution Algorithm with Ensemble of Parameters and Mutation Strategies**. Applied Soft Computing, v. 11, n. 2, p. 1679-1696, 2011.

MARINGANTI, C.; CHAUBEY, I.; ARABI, M.; ENGEL, B. **Cost Alternatives for NPS Pollution Control**. New York: Springer Science+Business Media, 2011.

MELO, W.A.X.; FAMPA, M.C.; RAUPP, F.M.P. **Evolução Diferencial Aperfeiçoada para Otimização Contínua Restrita**. XLII Simpósio Brasileiro de Pesquisa Operacional, Bento Gonçalves, Rio de Janeiro, 2010.

MIETTINEN, K. **Nonlinear Multiobjective Optimization**. 1.ed. New York: Springer Science+Business Media, 1998.

MIRJALILI, S. **Dragonfly Algorithm: A New Meta-Heuristic Optimization Technique for Solving Single-Objective, Discrete, and Multi-Objective Problems**. Neural Computing and Applications, p. 1-21, 2015.

MIRJALILI, S. **The Ant Lion Optimizer**. Advances in Engineering Software, v. 83, p. 80-98, 2015.

MIRJALILI, S.; MIRJALILI, S. M. **Multi-Verse Optimizer: A Nature-Inspired Algorithm for Global Optimization**. Neural Computing & Applications, v. 27, 495-513, 2016.

MIRJALILI, S.; MIRJALILI, S. M.; LEWIS, A. **Grey Wolf Optimizer**. Advances in Engineering Software, v. 69, p. 46-61, 2014.

QIN, A. K.; HUANG, V. L.; SUGANTHAN, P. N. **Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization**. IEEE Transactions on Evolutionary Computation, v. 13, n. 2, p. 398-417, 2009.

QUINZANI, C.M. **Otimização Multiobjetivo de Portfolios utilizando Algoritmos Evolutivos**. 2010. 50 f. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP, 2010.

OLIVEIRA, A.C.M. **Algoritmos Evolutivos para Problemas de Otimização Numérica com Restrições**. Laboratório Associado de Computação e Matemática Aplicada, Instituto Nacional de Pesquisas Espaciais, São Jose dos Campos, SP.

OLIVEIRA, G.T.S. **Estudos e Aplicações da Evolução Diferencial**. 2006. 95 f. Dissertação (Mestre em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, MG, 2006.

PAIVA, A.L.O. **Aplicação do Método de Evolução Diferencial à Otimização de um Ciclo de Refrigeração por Compressão de Vapor de Dois Estágios Através da Análise Exérgica**. 2001. 94 f. Dissertação (Mestre em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2001.

PRICE, K. **Genetic Annealing**, Dr. Dobb's Journal, p. 127-132, 1994.

RAM, V. D.; CHIDAMBARAM, M. **Simple Method of Designing Centralized PI Controllers for Multivariable Systems Based on SSGM**. ISA Transactions, v. 56, p. 252-260, 2015.

RAMPAZZO, P.C.B.; YAMAKAMI, A.; FRANÇA, F.O. Algoritmo Genético e Evolução Diferencial para a Resolução do Problema de Planejamento Hidrelétrico. In:_____. **Meta-Heurísticas em Pesquisa Operacional**, Universidade Estadual de Campinas, Engenharia Elétrica, Campinas, SP, 2014.

REYNOSO-MEZA, G. **Controller Tuning by Means of Evolutionary Multiobjective Optimization: A Holistic Multiobjective Optimization Design Procedure**. España: Universitat Politècnica de València, Valencia, 2014.

REYNOSO-MEZA, G.; BLASCO, X.; SANCHIS, J. MARTÍNEZ, M. **Controller Tuning Using Evolutionary Multi-Objective Optimization: Current Trends and Applications**. Control Engineering Practice, v. 28, p. 58-73, 2014.

SAHIB, M. A.; AHMED, B. S. **A New Multiobjective Performance Criterion Used in PID Tuning Optimization Algorithms**. Journal of Advanced Research, v. 7, n. 1, p. 125-134, 2016.

SAMPAIO, P.R. **Teoria, Métodos e Aplicações de Otimização Multiobjetivo**. 2011, 79 f. Dissertação (Mestre em Ciências da Computação) – Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, SP, 2011.

SAYED, M.; GHARGHORY, S. M.; KAMAL, H.A. **Gain Tuning PI Controllers for Boiler Turbine Unit a New Hybrid Jump PSO**. Journal of Electrical Systems and Information Technology, v. 2, n. 1, p. 99-110, 2015.

SEGUNDO, G. A. S. **Uma Abordagem Usando Evolução Diferencial para Solucionar Problemas de Programação em Dois Níveis Não Lineares**. 2012. 85 f. Dissertação (Mestre em Informática na área de Inteligência Computacional) – Departamento de Informática, Centro Tecnológico, Universidade Federal do Espírito Santo, Vitória, ES, 2012.

SIMON, D. **Evolutionary Optimization Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence**. Hoboken: John Wiley & Sons, Inc., 2013.

SOARES, G.L. **Algoritmos Genéticos: Estudo, Novas Técnicas e Aplicações**. 1997. 145 f. Tese (Doutorado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica, Centro de Pesquisa e Desenvolvimento em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, MG, 1997.

STORN, R. **Differential Evolution Research – Trends and Open Questions**. Germany: München, 1996.

STORN R.; PRICE, K. **Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Space**. Journal of Global Optimization, v.23, München, Germany, 1995.

SUTHA, S. LAKSHMI, P. SANKARANARAYANA, S. **Fractional-Order Sliding Mode Controller Design for a Modified Quadruple Tank Process via Multi-Level Switching**. Computer & Electrical Engineering, v. 45, p. 10-21, 2015.

TSAI, J.T.; HO, W.H.; CHOU, J.H.; GUO, C.Y. **Optimal Approximation of Linear Systems Using Taguchi-Sliding-Based Differential Evolution Algorithm**. Applied Soft Computing, v.11, n. 2, p. 2007-2016, 2011.

VELASCO, L. E. R; RAMÍREZ, O. A. D.; VEJA, V. P.; **Wavenet Fuzzy PID Controller for Nonlinear MIMO System: Experimental Validation on a High-End Haptic Robotic Interface**. Applied Soft Computing, v. 40, p. 199-205, 2016.

VLACHOS, C.; WILLIAMS, D.; GOMM, J.B. **Solution to the Shell Standard Control Problem Using Genetically Tuned PID Controllers**. Control Engineering Practice, v. 10, n. 2, p. 151-163, 2002.

VISALAKSHI, S.; BASKAR, S. **Covariance Matrix Adapted Evolution Strategy – Based Decentralized Congestion Management for Multilateral Transaction**. IET Generation, Transmission & Distribution, v. 4, n. 3, p. 400-417, 2010.

TANG, Y.; GAO, H.; DU, W.; LU, J. **Robust Multiobjective Controllability of Complex Neural Networks**. IEEE/ACM Transactions on Computational Biology and Bioinformatics, v. 13, n.4, p. 778-791, 2015.

XU, G.; LIU, Y.; GAO, Q. **Multi-Objective Optimization of a Continuous Bio-Dissimilation Process of Glycerol to, 1, 3-Propanediol**. Journal of Biotechnology, v. 219, p. 59-71, 2016.

WALPOLE, R. E.; MYERS, R. H.; MYERS, S. L.; YE, K. **Probability and Statistics for Engineers and Scientists**. 9th ed. USA: Prentice Hall, 2011.

WANG, Y.; CAI, Z.; ZHANG, Q. **Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters**. IEEE Transactions on Evolutionary Computation, v. 14, n. 1, p. 55-66, 2011.

YANG, Q.; ZHU, J.; XU, X.; LU, J. **Simultaneous Control of Indoor Air Temperature and Humidity for a Chilled Water Based Air Conditioning System Using Neural Networks**. Energy and Buildings, v. 110, p. 159-169, 2016.

YILDIZ, A.R; **A New Hybrid Differential Evolution Algorithm for the Selection of Optimal Machining Parameters in Milling Operations**. Applied Soft Computing, v. 13, n. 3, p. 1561-1566, 2013.

ZHANG, J.; SANDERSON, A. C.; **JADE: Adaptive Differential Evolution with Optional External Archive**. IEEE Transactions on Evolutionary Computation, v. 13, n. 5, p. 945-958, 2009.

ZHANG, Y.H.;GONG, Y.J.; CHEN, W. N.; ZHANG, J. **Composite Differential Evolution with Queueing Selection for Multimodal Optimization**. IEEE Congress on Evolutionary Computation, v. 1, p. 425-432, Sendai, Japão, 2015.

ZITZLER, E; THIELE, E.Z. **Multiobjective Evolutionary Algorithms: A comparative Case Study and the Strength Pareto Approach**. IEEE Transactions on Evolutionary Computation, v. 3, n. 4, p. 257-271, 1999.

ZHU, W.; TANG, Y.; FANG, J.; ZHANG W. **Adaptive Population Tuning Scheme for Differential Evolution**. Information Sciences, v. 223, p. 164-191, 2013.

APÊNDICES

APÊNDICE 1 - COMPARATIVO DOS ALGORITMOS POR MEIO DO TESTE T E TESTE DE WILCOXON

Neste apêndice são exibidas as tabelas de comparativo do Teste T e teste Wilcoxon entre todos os algoritmos.

A TABELA A.1 detalha o Teste T entre os algoritmos derivados do DE para o *boiler turbine*.

A TABELA A.2 mostra o Teste T entre os algoritmos baseados em população e também entre o clássico DE para o *boiler turbine*.

A TABELA A.3 apresenta o Teste Wilcoxon entre os algoritmos derivados do DE para o *quadruple tank*.

A TABELA A.4 mostra o Teste de Wilcoxon entre os algoritmos baseados em população e também entre o clássico DE para o *quadruple tank*.

TABELA A.1 - TESTE T – ALGORITMOS DERIVADOS DO DE – *BOILER TURBINE*

		CMAES	CODE	DE	EPSDE	JADE	jDE	SaDE
CMAES versus	<i>p-value</i>	-	5,1825E-011	5,0979E-11	5,0963E-11	5,3004E-11	5,2697e-11	5,9992E-11
	h	-	1	1	1	1	1	1
CODE versus	<i>p-value</i>	5,1825E-11	-	1,3043E-04	7,7896E-05	0,0131	0,0166	1,5029E-12
	h	1	-	1	1	1	1	1
DE versus	<i>p-value</i>	5,0979E-11	1,3043E-04	-	2,7416E-07	2,5716E-04	5,3611E-05	5,6754E-13
	h	1	1	-	1	1	1	1
EPSDE versus	<i>p-value</i>	5,0963E-11	7,7896E-05	2,7416E-07	-	2,1507E-04	4,0723E-05	2,2485E-13
	h	1	1	1	-	1	1	1
JADE versus	<i>p-value</i>	5,3004E-11	0,0131	2,5716E-04	2,1507E-04	-	0,4847	4,9031E-04
	h	1	1	1	1	-	0	1
jDE versus	<i>p-value</i>	5,2697e-11	0,0166	5,3611E-05	4,0723E-05	0,4847	-	5,1499e-06
	h	1	1	1	1	0	-	1
SaDE versus	<i>p-value</i>	5,9992E-11	1,5029E-12	5,6754E-13	2,2485E-13	4,9031E-04	5,1499e-06	-
	h	1	1	1	1	1	1	-

FONTE: DO AUTOR (2016)

TABELA A.2 - TESTE T – ALGORITMOS BASEADOS EM POPULAÇÃO – BOILER TURBINE

		ALO	DE	DA	GWO	MVO	PSO
ALO versus	<i>p-value</i>	-	4,47E-06	0,93	8,2E-05	2,53E-05	4,16E-06
	h	-	1	0	1	1	1
DE versus	<i>p-value</i>	4,47E-06	-	9,53E-07	2,27E-04	1,77E-05	0,11
	h	1	-	1	1	1	0
DA versus	<i>p-value</i>	0,93	9,53E-07	-	1,15E-05	2,55E-05	1,01E-06
	h	0	1	-	1	1	1
GWO versus	<i>p-value</i>	8,20E-05	2,27E-04	1,15E-05	-	1,89E-05	6,16E-04
	h	1	1	1	-	1	1
MVO versus	<i>p-value</i>	2,53E-05	1,77E-05	2,55E-05	1,89E-05	-	1,80E-05
	h	1	1	1	1	-	1
PSO versus	<i>p-value</i>	4,16E-06	0,11	1,01E-06	6,16E-04	1,80E-05	-
	h	1	0	1	1	1	-

FONTE: DO AUTOR (2016)

TABELA A.3 - TESTE WILCOXON – ALGORITMOS DERIVADOS DO DE – QUADRUPLE TANK

		CMAES	CODE	DE	EPSDE	JADE	jDE	SaDE
CMAES versus	<i>p-value</i>	-	0,05	0,10	0,04	0,08	0,0760	0,06
	h	-	0	0	0	0	0	0
CODE versus	<i>p-value</i>	0,05	-	6,12E-10	0,94	2,1E-03	3,91E-05	0,62
	h	0	-	1	0	1	1	0
DE versus	<i>p-value</i>	0,10	6,12E-10	-	1,27E-16	0,15	5,9E-03	8,95E-11
	h	0	1	-	1	0	1	1
EPSDE versus	<i>p-value</i>	0,04	0,94	1,27E-16	-	4,5E-03	6,42E-12	0,10
	h	0	0	1	-	1	1	0
JADE versus	<i>p-value</i>	0,08	2,1E-03	0,15	4,5E-03	-	0,88	1,5E-03
	h	0	1	0	1	-	0	1
jDE versus	<i>p-value</i>	0,07	3,91E-05	5,9E-03	6,42E-12	0,88	-	4,58E-07
	h	0	1	1	1	0	-	1
SaDE versus	<i>p-value</i>	0,06	0,62	8,95E-11	0,10	1,5E-03	4,58E-07	-
	h	0	0	1	0	1	1	-

FONTE: DO AUTOR (2016)

TABELA A.4 - TESTE WILCOXON – ALGORITMOS BASEADOS EM POPULAÇÃO – QUADRUPLE TANK

		ALO	DA	DE	GWO	MVO	PSO
ALO versus	<i>p-value</i>	-	0,63	1,14E-11	1,66E-13	3,10E-12	1,20E-16
	h	-	0	1	1	1	1
DE versus	<i>p-value</i>	1,14E-11	8,9E-03	-	4,5E-03	3,98E-15	1,20E-16
	h	1	1	-	1	1	1
DA versus	<i>p-value</i>	0,63	-	8,9E-03	0,02	1,28E-13	7,06E-18
	h	0	-	1	1	1	1
GWO versus	<i>p-value</i>	1,66E-13	0,02	4,5E-03	-	3,20E-12	7,06E-18
	h	1	1	1	-	1	1
MVO versus	<i>p-value</i>	3,10E-12	1,28E-13	3,98E-15	3,20E-12	-	7,06E-18
	h	1	1	1	1	-	1
PSO versus	<i>p-value</i>	1,20E-16	7,06E-18	1,20E-16	7,06E-18	7,06E-18	-
	h	1	1	1	1	1	-

FONTE: DO AUTOR (2016)

APÊNDICE 2 – COMPARAÇÃO ENTRE OS MELHORES RESULTADOS - GRÁFICO DA MÍNIMA, MÁXIMA E MEDIANA

Este apêndice apresenta a comparação (mínima, máxima e mediana) entre o clássico DE, o melhor algoritmo derivado do DE e o melhor algoritmo baseado em população.

A FIGURA A2.0.1 mostra o comparativo entre o clássico DE, EPDSE e o PSO para o *boiler turbine*, pois estes apresentaram os melhores resultados quando comparados a outros algoritmos descritos nesta dissertação. Nota-se que o algoritmo com o melhor desempenho foi o EPSDE.

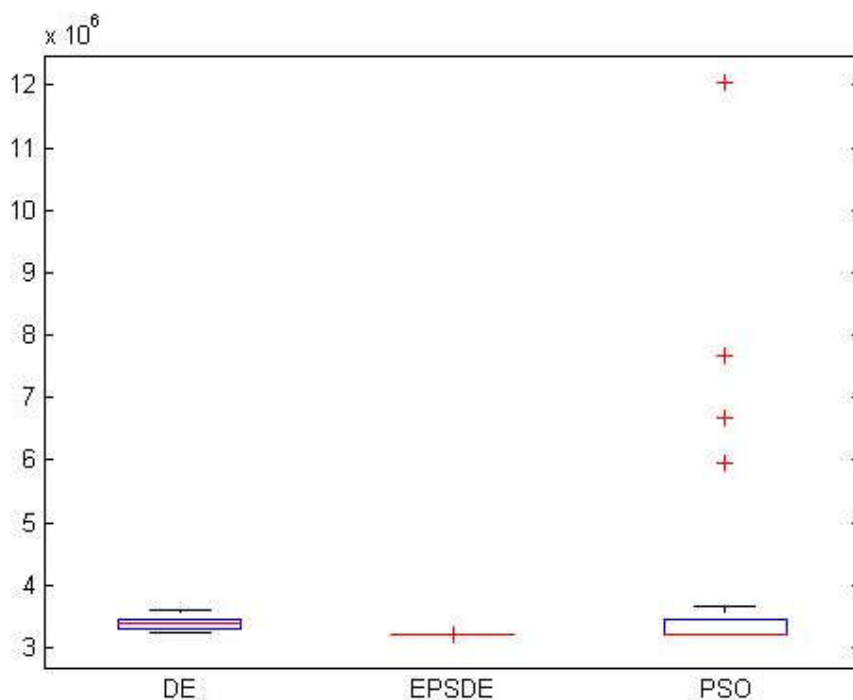


FIGURA A2.0.1 - COMPARAÇÃO ENTRE OS MELHORES RESULTADOS PARA *BOILER TURBINE*
 FONTE: DO AUTOR (2016)

A Figura A2.0.2 mostra o comparativo entre o clássico DE, EPDSE e o PSO para o *quadruple tank*, pois estes apresentaram os melhores resultados quando comparados a outros algoritmos descritos nesta dissertação. Nota-se pelos resultados obtidos que o algoritmo com o melhor desempenho foi o EPSDE.

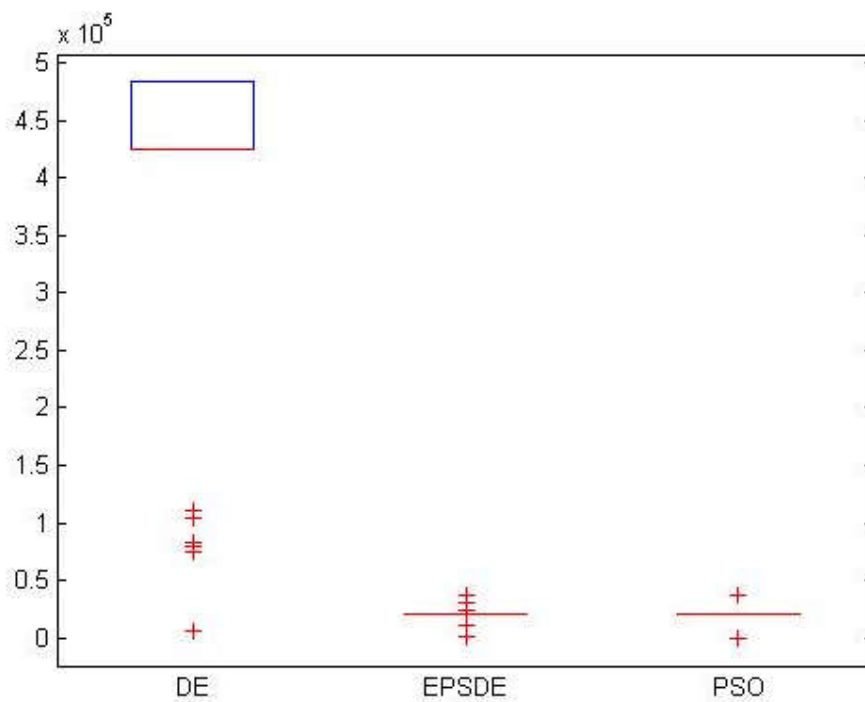


FIGURA A2.0.2 - COMPARAÇÃO ENTRE OS MELHORES RESULTADOS PARA *QUADRUPLE TANK*
FONTE: DOA AUTOR (2016)