

**UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE TECNOLOGIA - ITEC  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**CARLOS ALBERTO OLIVEIRA DE FREITAS**

**Algoritmo Memético Cultural para Otimização de Problemas de Variáveis  
Reais**

TD 05/2019

UFPA / ITEC / PPGEE  
Campus Universitário do Guamá  
Belém-Pará-Brasil  
2019

**UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE TECNOLOGIA - ITEC  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**CARLOS ALBERTO OLIVEIRA DE FREITAS**

**Algoritmo Memético Cultural para Otimização de Problemas de Variáveis  
Reais**

Tese submetida a Banca Examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA para a obtenção do Grau de Doutor em Engenharia Elétrica na área de Computação Aplicada.

UFPA / ITEC / PPGEE  
Campus Universitário do Guamá  
Belém-Pará-Brasil

2019

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD  
Sistema de Bibliotecas da Universidade Federal do Pará  
Gerada automaticamente pelo módulo Ficcat, mediante os dados fornecidos pelo(a) autor(a)

---

F862a Freitas, Carlos Alberto Oliveira de  
Algoritmo Memético Cultural para Otimização de Problemas de  
Variáveis Reais / Carlos Alberto Oliveira de Freitas. — 2019.  
xvi, 109 f. : il. color.

Orientador(a): Prof. Dr. Roberto Célio Limão de Oliveira  
Coorientador(a): Prof. Dr. Deam James Azevedo da Silva  
Tese (Doutorado) - Programa de Pós-Graduação em Engenharia  
Elétrica, Instituto de Tecnologia, Universidade Federal do Pará,  
Belém, 2019.

1. Algoritmos Culturais. 2. Algoritmos Meméticos. 3. Busca  
Local. 4. Otimização multimodal. 5. Problemas de otimização  
com restrições. I. Título.

CDD 620

---

UNIVERSIDADE FEDERAL DO PARÁ  
 INSTITUTO DE TECNOLOGIA  
 PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**“ALGORITMO MEMÉTICO CULTURAL PARA OTIMIZAÇÃO DE  
 PROBLEMAS DE VARIÁVEIS REAIS”**

AUTOR: CARLOS ALBERTO OLIVEIRA DE FREITAS

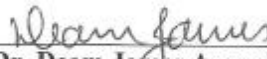
TESE DE DOUTORADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO  
 COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO  
 JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE DOUTOR EM ENGENHARIA  
 ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA.

APROVADA EM: 29/03/2019

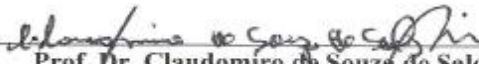
**BANCA EXAMINADORA:**



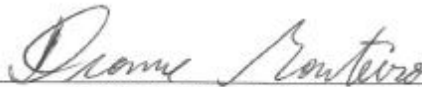
\_\_\_\_\_  
**Prof. Dr. Roberto Célio Limão de Oliveira**  
 (Orientador – PPGEE/UFPA)



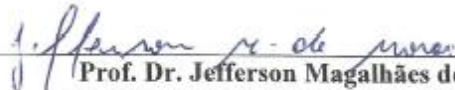
\_\_\_\_\_  
**Prof. Dr. Deam James Azevedo da Silva**  
 (Co-Orientador – UFOPA)



\_\_\_\_\_  
**Prof. Dr. Claudomiro de Souza de Sales Junior**  
 (Avaliador Interno – PPGEE/UFPA)



\_\_\_\_\_  
**Prof. Dr. Dionne Cavalcante Monteiro**  
 (Avaliador Externo ao Programa – ICEN/UFPA)



\_\_\_\_\_  
**Prof. Dr. Jefferson Magalhães de Moraes**  
 (Avaliador Externo ao Programa – ICEN/UFPA)

\_\_\_\_\_  
**Prof. Dr. Wagner Figueiredo Sacco**  
 (Avaliador Externo – UFOPA)

**VISTO:**

\_\_\_\_\_  
**Prof.ª Dr.ª Maria Emilia de Lima Testes**  
 (Coordenadora do PPGEE/ITEC/UFPA)

## AGRADECIMENTOS

A Deus por tudo, pois apesar de minhas falhas tenho recebido muitas graças.

Ao meu pai Antônio Coelho de Freitas (in memoriam), ao meu tio/pai Expedito Euclides Pereira (in memoriam) as minhas mães Damiana, Maria de Lourdes, Maria José e Maria Neuza (in memoriam), pelos ensinamentos de honestidade, humildade, perseverança e educação, e por me conduzirem na busca do conhecimento.

Aos meus irmãos: Paulo, Franco e Frâncio e minha irmã Ivanete, por estarem sempre presentes na minha vida e por terem orgulho de mim.

A minha esposa Ereunice Costa de Freitas (Nice), por me amar, compreender, incentivar e organizar minha vida.

As minhas cunhadas Cléo, Diva, Dilma e Simone, por me ajudarem e contribuir com incentivo e apoio nesta caminhada.

As minhas filhas Lívia e Gabriela, meus filhos Pedro, Gustavo e Carlos Jr., pelo incentivo e compreensão dos momentos que tive que me dedicar aos estudos e não os acompanhar em outras atividades.

Ao meu Orientador, Professor Dr. Roberto Célio Limão de Oliveira, e ao meu Co-orientador Professor Dr. Deam James Azevedo da Silva, pela paciência, confiança, pelo apoio nos momentos que quis me desviar, mas suas orientações me trouxeram sempre para o caminho correto.

Aos amigos Edson Farias, Haroldo Melo e João Carlos, que estiveram comigo desde o mestrado até esta conquista, sempre com muita motivação e companheirismo.

Agradeço ao meu amigo Jandecy Cabral e Tereza Felipe, pelo suporte, confiança, incentivo nesta caminhada proporcionada pelo ITEGAM, representado por estes dois seres de luz.

Aos meus amigos de turma do curso de doutorado e do ITEGAM, Jorge Almeida Brito Júnior, Manoel Henrique Reis Nascimento, David Barbosa de Alencar, Nadime Mustafa, Milton Fonseca Júnior e outros que fizeram parte da caminhada, também aos colegas e amigos conquistados neste curso.

Aos colegas Paulo, Alarico e Lene do ITEGAM, que sempre torceram para que tivesse mais essa vitória na minha vida.

Agradeço aos professores do programa de pós-graduação em engenharia elétrica do ITEC-UFPA, e todos os outros que me guiaram por esse novo caminho.

Ao Instituto de Tecnologia José Rocha Sérgio Cardoso e ao Sr. Ilídio Costa, incentivador na aquisição de conhecimentos e novas tecnologias.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>1</b>
<b>1.1</b>	<b>Considerações Iniciais.....</b>	<b>1</b>
1.1.1	Motivação.....	3
1.1.2	Objetivos da Tese .....	4
<b>1.2</b>	<b>O Estado da Arte .....</b>	<b>5</b>
1.2.1	<i>A two-phase tabu-evolutionary algorithm for the 0-1 multidimensional knapsack problem.....</i>	<i>7</i>
1.2.2	<i>The Bounded Beam Search algorithm for the Block Relocation Problem .....</i>	<i>8</i>
1.2.3	<i>Developing a Dynamic Neighborhood Structure for an Adaptive Hybrid Simulated Annealing – Tabu Search Algorithm to Solve the Symmetrical Traveling Salesman Problem..</i>	<i>8</i>
1.2.4	<i>A novel class of niche hybrid Cultural Algorithms for continuous engineering optimization .....</i>	<i>9</i>
1.2.5	<i>Opposition-based Memetic Search for the Maximum Diversity Problem.....</i>	<i>10</i>
1.2.6	<i>Path-relinking Tabu search for the multi-objective flexible job shop scheduling problem.....</i>	<i>11</i>
1.2.7	<i>Balancing search direction in cultural algorithm for enhanced global numerical optimization .....</i>	<i>11</i>
1.2.8	<i>Energy and Labor Aware Production Scheduling for Industrial Demand Response Using Adaptive Multi-Objective Memetic Algorithm .....</i>	<i>12</i>
1.2.9	<i>A Tabu search based hybrid evolutionary algorithm for the max-cut problem .....</i>	<i>13</i>
1.2.10	<i>Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, Instances, and Algorithm .....</i>	<i>13</i>
<b>1.3</b>	<b>Justificativa .....</b>	<b>15</b>
<b>1.4</b>	<b>Estrutura do Trabalho.....</b>	<b>15</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>16</b>
<b>2.1</b>	<b>Algoritmos Evolutivo .....</b>	<b>16</b>
<b>2.2</b>	<b>Algoritmo Genético (AG).....</b>	<b>17</b>
2.2.1	Considerações Iniciais.....	17
2.2.2	Características dos Algoritmos Genéticos (AGs).....	18
2.2.3	Algoritmo Genético Simples .....	19

<b>2.3</b>	<b>Algoritmo Cultural (AC)</b> .....	<b>21</b>
2.3.1	Considerações Iniciais .....	21
2.3.2	Características culturais .....	21
2.3.3	Funcionamento de um Algoritmo Cultural .....	21
2.3.4	Características do Algoritmo Cultural .....	23
2.3.5	Espaço Populacional .....	23
2.3.6	Espaço de Crenças .....	24
2.3.7	Protocolos de Comunicação .....	25
2.3.8	Aplicação do Algoritmo Cultural .....	25
<b>2.4</b>	<b>Estratégias de Busca local</b> .....	<b>26</b>
2.4.1	Heurísticas e Meta-heurísticas .....	26
2.4.2	<i>Hill Climbing</i> .....	27
2.4.3	<i>Simulated Annealing (SA)</i> .....	28
2.4.4	<i>Tabu Search (TS)</i> .....	30
2.4.5	<i>Beam Search (BS)</i> .....	32
<b>2.5</b>	<b>Abordagem Memética</b> .....	<b>33</b>
2.5.1	Considerações Iniciais .....	33
2.5.2	Implementação de Algoritmos Meméticos .....	33
2.5.3	Aplicação do Algoritmo Memético .....	35
<b>2.6</b>	<b>Técnicas para medidas de desempenho de algoritmos</b> .....	<b>37</b>
2.6.1	Métodos paramétricos .....	38
2.6.2	Métodos não paramétricos .....	39
2.6.3	Teste de <i>Friedman</i> .....	39
2.6.4	Teste de <i>Friedman Aligned</i> .....	40
2.6.5	Teste de <i>Quade</i> .....	40
2.6.6	<i>Hellinger-TOPSIS</i> .....	41
<b>3</b>	<b>METODOLOGIA APLICADA</b> .....	<b>44</b>
<b>3.1</b>	<b>Pesquisa bibliográfica</b> .....	<b>44</b>
<b>3.2</b>	<b>Desenvolvimento do algoritmo memético</b> .....	<b>45</b>
<b>3.3</b>	<b>Avaliação do algoritmo desenvolvido na aplicação de funções de benchmark</b> ..	<b>45</b>
3.3.1	Funções de <i>benchmark</i> .....	45
3.3.2	Avaliação pela técnica <i>Hellinger-TOPSIS</i> .....	46
3.3.3	Avaliação pelas técnicas de <i>Friedman</i> , <i>Friedman Aligned</i> e <i>Quade</i> .....	48

<b>3.4</b>	<b>Avaliação do algoritmo memético na aplicação em problemas de engenharia ..</b>	<b>50</b>
3.4.1	Engenharia Elétrica .....	50
3.4.2	Engenharia Mecânica .....	56
3.4.3	Engenharia Civil.....	59
<b>4</b>	<b>ALGORITMO MEMÉTICO CULTURAL.....</b>	<b>61</b>
<b>4.1</b>	<b>Considerações Iniciais.....</b>	<b>61</b>
<b>4.2</b>	<b>Melhorias no Algoritmo Cultural .....</b>	<b>61</b>
<b>4.3</b>	<b>Hibridização do Algoritmo Cultural com a Busca Local .....</b>	<b>64</b>
4.3.1	Algoritmo Cultural com Busca Tabu (AC+BT).....	65
4.3.2	Algoritmo Cultural com Hill Climbing (AC+HC).....	65
4.3.3	Algoritmo Cultural com Simulated Annealing (AC+SA).....	66
4.3.4	Algoritmo Cultural com Beam Search (AC+BS).....	66
<b>5</b>	<b>RESULTADOS.....</b>	<b>67</b>
<b>5.1</b>	<b>Introdução.....</b>	<b>67</b>
<b>5.2</b>	<b>Resultados para otimização de funções multimodais <i>benchmark</i> .....</b>	<b>67</b>
5.2.1	Resultados e discussões ( <i>Hellinger-TOPSIS</i> ) .....	67
5.2.2	Resultados e discussões ( <i>Friedman, Friedman Aligned e Quade</i> ).....	74
<b>5.3</b>	<b>Aplicação na solução de problemas de otimização em engenharia .....</b>	<b>76</b>
5.3.1	Análise dos resultados (Engenharia elétrica) .....	76
5.3.2	Engenharia Mecânica .....	88
5.3.3	Engenharia Civil.....	90
<b>6</b>	<b>CONCLUSÃO E RECOMENDAÇÕES PARA TRABALHOS FUTUROS.....</b>	<b>91</b>
<b>6.1</b>	<b>Conclusão .....</b>	<b>91</b>
<b>6.2</b>	<b>Recomendações para trabalhos futuros .....</b>	<b>92</b>
<b>6.3</b>	<b>Publicações geradas pela tese .....</b>	<b>93</b>
<b>7</b>	<b>REFERÊNCIAS .....</b>	<b>94</b>
<b>8</b>	<b>ANEXOS .....</b>	<b>106</b>



## LISTA DE FIGURAS

<b>Figura 1.1:</b>	(a) Solução e (b) Representação. ....	14
<b>Figura 2.1:</b>	Algoritmo Básico de um AG. ....	19
<b>Figura 2.2:</b>	Fluxograma de um Algoritmo Genético Básico. ....	20
<b>Figura 2.3:</b>	Fluxograma do Algoritmo Cultural. ....	22
<b>Figura 2.4:</b>	Algoritmo básico <i>Hill Climbing</i> .....	28
<b>Figura 2.5:</b>	Processo de recozimento.....	29
<b>Figura 2.6:</b>	Algoritmo do SA.....	30
<b>Figura 2.7:</b>	Algoritmo básico Tabu. ....	32
<b>Figura 2.8:</b>	Exemplo de tabela para a matriz de decisão. ....	42
<b>Figura 3.1:</b>	Organograma para o teste. ....	48
<b>Figura 3.2:</b>	Organograma para o teste. ....	49
<b>Figura 3.3:</b>	O fluxograma do método. ....	55
<b>Figura 3.4:</b>	A mola helicoidal de compressão. ....	57
<b>Figura 3.5:</b>	Vaso de pressão.....	57
<b>Figura 3.6:</b>	Redutor de velocidade.....	58
<b>Figura 3.7:</b>	Coluna Tubular uniforme.....	59
<b>Figura 4.1:</b>	Algoritmo cultural com destaque nos conhecimentos deste trabalho. ....	62
<b>Figura 4.2:</b>	Áreas separadas de melhores resultados da busca local. ....	62
<b>Figura 4.3:</b>	Áreas congruentes de melhores resultados da busca local.....	63
<b>Figura 4.4:</b>	Pseudocódigo do algoritmo proposto.....	64
<b>Figura 4.5:</b>	Topologia de espaço de estados unidimensional. ....	65
<b>Figura 4.6:</b>	Estrutura da busca em feixes com $k=2$ . ....	66
<b>Figura 5.1:</b>	Função 1 no cenário 1. ....	71
<b>Figura 5.2:</b>	Função 1 no cenário 2. ....	72
<b>Figura 5.3:</b>	Função 1 no cenário 1. ....	72
<b>Figura 5.4:</b>	Função 1 no cenário 2. ....	73
<b>Figura 5.5:</b>	Função de Custos e função de emissões vs Gerações, usando todos os motores.....	80
<b>Figura 5.6:</b>	Função de Custos e função de emissões vs Gerações, desligando os motores com maior custo incremental.....	81
<b>Figura 5.7:</b>	Frente de Pareto dos custos versus emissões usando todos os motores.....	81
<b>Figura 5.8:</b>	Frente de Pareto desligando os motores com maior custo incremental. ....	82

<b>Figura 5.9:</b>	Convergência AG vs AC. ....	84
<b>Figura 5.10:</b>	Convergência AC, AC+BT e AC+SA. ....	84
<b>Figura 5.11:</b>	Curva característica de convergência.....	87
<b>Figura 8.1:</b>	Estrutura principal do JEF Customizada.....	107
<b>Figura 8.2:</b>	Classes de modelagem dos problemas. ....	108
<b>Figura 8.3:</b>	Código fonte da função F1.....	109

## LISTA DE TABELAS

<b>Tabela 1.1:</b>	Processo físico de recozimento versus o algoritmo de recozimento simulado. ...6
<b>Tabela 2.1:</b>	Processo físico de recozimento versus o algoritmo de recozimento simulado. 29
<b>Tabela 2.2:</b>	Testes estatísticos paramétricos e não-paramétricos..... 38
<b>Tabela 3.1:</b>	Funções Básicas de benchmark. .... 46
<b>Tabela 3.2:</b>	Funções Híbridas de benchmark. .... 46
<b>Tabela 3.3:</b>	Cenário 1..... 46
<b>Tabela 3.4:</b>	Cenário 2..... 47
<b>Tabela 3.5:</b>	Algoritmos para teste. .... 47
<b>Tabela 3.6:</b>	Cenário 1 e 2 – Funções Básicas e Híbridas. .... 49
<b>Tabela 3.7:</b>	Dados característicos dos geradores do estudo de caso. .... 53
<b>Tabela 3.8:</b>	Matriz de emissões dos geradores do estudo de caso. .... 53
<b>Tabela 3.9:</b>	Matriz de perdas dos geradores da usina (Valores devem ser multiplicados por 1e-4)..... 53
<b>Tabela 3.10:</b>	Dados dos geradores do Sistema de Teste IEEE de 13 Unidades..... 55
<b>Tabela 3.11:</b>	Matriz de Emissões do Sistema de Teste IEEE de 13 Unidades. .... 56
<b>Tabela 3.12:</b>	Dados para o projeto. .... 60
<b>Tabela 5.1:</b>	Matriz de Decisão das médias para o cenário 1. .... 67
<b>Tabela 5.2:</b>	Matriz de Decisão do desvio padrão para o cenário 1. .... 68
<b>Tabela 5.3:</b>	Matriz de Decisão das médias para o cenário 2. .... 68
<b>Tabela 5.4:</b>	Matriz de Decisão do desvio padrão para o cenário 2. .... 69
<b>Tabela 5.5:</b>	PIS e NIS para as médias de cada critério da Matriz de Decisão para o cenário 1..... 69
<b>Tabela 5.6:</b>	PIS e NIS para o Desvio Padrão cada critério da Matriz de Decisão para o cenário 1..... 69
<b>Tabela 5.7:</b>	PIS e NIS para as médias de cada critério da Matriz de Decisão para o cenário 2..... 69
<b>Tabela 5.8:</b>	PIS e NIS para o Desvio Padrão cada critério da Matriz de Decisão para o cenário 2..... 69
<b>Tabela 5.9:</b>	Resultado para o cenário 1. .... 70
<b>Tabela 5.10:</b>	Resultado para o cenário 2..... 70
<b>Tabela 5.11:</b>	Entrada de dados de D-10. .... 74
<b>Tabela 5.12:</b>	Entrada de dados de D-30. .... 74

<b>Tabela 5.13:</b> Resultado para D-10. ....	75
<b>Tabela 5.14:</b> Resultado para D-30. ....	75
<b>Tabela 5.15:</b> Resultado para a junção de D-10 e D-30. ....	76
<b>Tabela 5.16:</b> Relatório final do comparativo multi-objetivo AC e SA (Energia gerada por gerador).....	77
<b>Tabela 5.17:</b> Comparativo multi-objetivo AC com Híbrido (Energia gerada por gerador)...	78
<b>Tabela 5.18:</b> Comparativo multi-objetivo AC com Híbrido (Emissão por gerador). ....	78
<b>Tabela 5.19:</b> Comparativo multi-objetivo AC com Híbrido (Custo por gerador). ....	79
<b>Tabela 5.20:</b> SA Clássico <i>versus</i> Algoritmos deste trabalho. ....	79
<b>Tabela 5.21:</b> DE Clássico <i>versus</i> Algoritmo deste trabalho.....	80
<b>Tabela 5.22:</b> Comparação de AC, AC+SA, AC+BT com o AG clássico (Custos). ....	83
<b>Tabela 5.23:</b> Comparação de AC, AC+SA, AC+BT com o AG clássico (Emissões). ....	83
<b>Tabela 5.24:</b> Comparação estatística para os 10 geradores.....	85
<b>Tabela 5.25:</b> Comparação de AC+SA, AC+BT com SA+BT (Custos).....	85
<b>Tabela 5.26:</b> Comparação de AC+SA, AC+BT com SA+BT (Emissões).....	85
<b>Tabela 5.27:</b> Resultados deste trabalho aplicado sistema de teste IEEE com 13 geradores. .	86
<b>Tabela 5.28:</b> Algoritmo proposto vs outros. ....	86
<b>Tabela 5.29:</b> Comparação estatística dos dados.....	87
<b>Tabela 5.30:</b> Melhores soluções para a mola de compressão helicoidal.....	88
<b>Tabela 5.31:</b> Comparação do AC+BT com outros algoritmos (melhor resultado encontrado para o Problema do Recipiente de Pressão) .....	89
<b>Tabela 5.32:</b> Resultados da otimização do redutor de velocidade. ....	89
<b>Tabela 5.33:</b> Melhores soluções para o exemplo da coluna tubular. ....	90

## LISTA DE ABREVIATURAS E SIGLAS

<b>LS</b>	<i>Local Search</i>
<b>TS</b>	<i>Tabu Search</i>
<b>BS</b>	<i>Beam Search</i>
<b>SA</b>	<i>Simulated Annealing</i>
<b>HC</b>	<i>Hill Climbing</i>
<b>AC</b>	Algoritmo Cultural
<b>AG</b>	Algoritmo Genético
<b>AM</b>	Algoritmo Memético

## RESUMO

A tecnologia deu grandes passos nos últimos anos, mas os recursos de computação para certas aplicações precisam de otimização para que os custos envolvidos na solução de alguns problemas não sejam altos. Existe uma área muito ampla de pesquisa para o desenvolvimento de algoritmos eficientes para problemas de otimização multimodal. Nas duas últimas décadas o uso de algoritmos evolutivos em otimização multimodal tem demonstrado ser um sucesso. Dentre esses algoritmos evolutivos, que são algoritmo de busca global, pode-se citar o uso dos Algoritmos Culturais. Um aprimoramento natural do Algoritmo Cultural é a sua hibridização com algum outro algoritmo de busca local, de forma a ter as vantagens da busca global combinada com a busca local. Entretanto os Algoritmos Culturais com busca local usados para otimização multimodal nem sempre são avaliados por testes estatísticos eficientes. O objetivo deste trabalho é analisar o comportamento do Algoritmo Cultural, com populações evoluídas pelo Algoritmo Genético, quando são utilizadas as heurísticas de busca locais: Busca Tabu, Busca de Feixe, Escalada e Recozimento Simulado. Uma das contribuições deste trabalho foi a atualização do conhecimento topográfico do algoritmo cultural pelo uso da área triangular definida pelos melhores resultados encontrados na busca local. Para realizar a análise, um algoritmo memético foi desenvolvido pela hibridização do algoritmo cultural com as heurísticas de busca local citadas, sendo selecionadas uma de cada vez. Os problemas do mundo real costumam ter características multimodais, então as avaliações foram realizadas usando funções de benchmark multimodais, que tiveram seus resultados avaliados por testes não paramétricos. Além disso, o algoritmo memético foi testado em problemas reais de otimização com restrições nas áreas de engenharia. Nas avaliações realizadas, o Algoritmo Cultural Memético desenvolvido apresentou melhores resultados quando comparado com os resultados disponíveis da literatura científica pesquisada.

**PALAVRAS-CHAVE:** Algoritmos Culturais, Algoritmos Meméticos, Busca Local, Problemas de otimização com restrições, Otimização multimodal.

## ABSTRACT

Technology has made great strides in recent years, but computing resources for certain applications need optimization so that the costs involved in solving some problems are not high. There is a very broad area of research for the development of efficient algorithms for multimodal optimization problems. In the last two decades the use of evolutionary algorithms in multimodal optimization has been shown to be a success. Among these evolutionary algorithms, which are global search algorithms, one can cite the use of Cultural Algorithms. A natural enhancement of the Cultural Algorithm is its hybridization with some other local search algorithm, so as to have the advantages of global search combined with local search. However, the local search Cultural Algorithms used for multimodal optimization are not always evaluated by efficient statistical tests. The objective of this work is to analyze the behavior of the Cultural Algorithm, with populations evolved by the Genetic Algorithm, when the local search heuristics are used: Tabu Search, Beam Search, Climbing and Simulated Annealing. One of the contributions of this work was the updating of the topographic knowledge of the cultural algorithm by the use of the triangular area defined by the best results found in the local search. To perform the analysis, a memetic algorithm was developed by hybridizing the cultural algorithm with the local search heuristics mentioned, being selected one at a time. Real world problems usually have multimodal characteristics, so the evaluations were performed using multimodal benchmark functions, which had their results evaluated by non-parametric tests. In addition, the memetic algorithm was tested on real optimization problems with constraints in the engineering areas. In the evaluations carried out, the developed Cultural Algorithm presented better results when compared to the available results of the researched scientific literature.

**KEYWORDS:** Cultural Algorithms, Memetics Algorithms, Local Search, Optimization Problems with Constraints, Multimodal Optimization.

# 1 INTRODUÇÃO

## *1.1 Considerações Iniciais*

A ideia de construção de heurísticas inspiradas em mecanismos baseados na adaptação dos seres vivos de acordo como ocorre na natureza, iniciou-se nas últimas quatro décadas do século XX. Embora se possa traçar suas raízes genealógicas desde os anos 1930, foi o surgimento da tecnologia de computação digital relativamente barata na década de 1960, que serviu como um importante catalisador para o campo (DE JONG, 2006). A evolução natural das espécies poderia ser encarada como um processo de aprender a se adaptar ao ambiente e otimizar a aptidão das espécies. Assim, poderíamos imitar o ponto de vista da genética moderna, ou seja, o princípio da "sobrevivência do mais apto", ao projetar algoritmos de otimização ou aprendizado (YU e GEN, 2010).

Na década de 1960, três grupos desenvolveram atividades que serviram para definir e moldar esta área. O primeiro grupo, trabalhou no uso de processos evolutivos na solução de problemas de otimização com parâmetros reais. Estas ideias desenvolvidas por Rechenberg e Schwefel, fizeram surgir a família de algoritmos de “estratégias evolutivas”. Um segundo grupo, que teve como precursor, Fogel, notou que através de técnicas evolutivas, existia o potencial de alcançar os objetivos da inteligência artificial. Uma estrutura evolutiva chamada de “programação evolucionária” foi desenvolvida com base em agentes inteligentes representados como máquinas de estado finitos. Nesta mesma década, Holland, observou que os processos evolutivos poderiam lidar com ambiente incertos e em mudança, pela implementação de sistemas adaptativos robustos. Esta visão levou a família inicial de “planos reprodutivos” que foram a base dos “algoritmos genéticos simples”.

Nos anos da década de 1970, uma parte das pesquisas nesta área, tentou por meio de estudos empíricos e extensões das teorias existentes, responder algumas questões deixadas



pelas especificações iniciais e análise desses simples algoritmos evolutivos (EAs). Ou seja, como caracterizar o comportamento de sistemas implementáveis? E como entender melhor como elas poderiam ser usadas para resolver problemas? Segundo DE JONG (2006), essas atividades resultaram no surgimento de três espécies distintas de EAs: programação evolutiva, estratégia evolutiva e algoritmos genéticos.

As atividades durante a década de 1980 resultaram em desenvolvimentos significativos em várias frentes na teoria e aplicação dos EAs. No final desta década e no início da década de 1990, apareceram diversas conferências de EA que possibilitaram a discussão e apresentação das teorias e aplicações. O efeito imediato foi um acordo sobre o termo "computação evolutiva" (EC) como o nome do campo e um compromisso para iniciar o primeiro jornal do campo, *Evolutionary Computation* (DE JONG, 2006). Em uma conferência no ano de 1994, Reynolds apresenta o algoritmo Cultural (AC) que é a base para desenvolvimento desta tese e será apresentado em uma seção própria.

Atualmente devido diversas circunstâncias, tais como, à complexidade computacional de resolver instâncias de problemas de grande escala (WANG *et al.*, 2014; WU *et al.*, 2015), a necessidade de melhorar agendamentos de acordo com critérios definidos em cada problema complexo (JIA e HU, 2014) e a busca do melhor dimensionamento de fontes alternativas de energia minimizando o custo da geração desta (KATSIKIANNIS e STAVRAKAKIS, 2014), busca-se cada vez mais formas de otimizar as atividades realizadas humanas.

Segundo LINDEN (2008), a otimização é uma ciência que está sempre em demanda, uma vez que se encontra direta ou indiretamente relacionada com capital e é empregada em todos os campos de aplicações tais como: engenharia civil, mecânica, automobilística, aérea, econômica, eletrônica, química, etc. O objetivo é encontrar boas soluções viáveis em uma escala de tempo aceitável mesmo que não exista garantia de que as melhores soluções possam ser encontradas (JIA e HU, 2014).

Os métodos de otimização modernos, também, por vezes, chamados métodos de otimização não tradicionais, surgiram como métodos populares para resolver problemas de otimização complexos de engenharia nos últimos anos (RAO, 2009). Sejam, no planejamento, no desenvolvimento ou nas verificações de cada etapa das atividades.

O uso da computação evolutiva na solução de problemas de otimização tem aumentado com a disponibilidade de recursos tecnológicos. Na computação evolutiva, há quatro paradigmas históricos que serviram como base para grande parte da atividade do campo: os algoritmos genéticos, a programação genética, as estratégias evolutivas e a programação evolutiva. As diferenças básicas entre esses paradigmas estão na natureza dos esquemas de

representação, os operadores de reprodução e métodos de seleção (ZHANG e KIM, 2000). A computação evolutiva oferece vantagens práticas para os pesquisadores que enfrentam problemas de otimização. Estas vantagens são várias, incluindo a simplicidade da abordagem, sua resposta robusta às diferentes circunstâncias, sua flexibilidade, e muitas outras (UMA *et al.*, 2011).

Um outro termo que apareceu no final dos anos 80, foi o termo algoritmos meméticos (AM), usado para denominar uma família de metaheurísticas que misturam vários conceitos de técnicas distintas, tais como, EAs e *Simulated Annealing* (SA). A filosofia central dos AMs, segundo MOSCATO *et al.* (2004) é a melhoria individual, mais cooperação e competição da população, pois estão presentes em muitos sistemas sociais / culturais. Os AMs muitas vezes são usados com outras denominações, as mais utilizadas são: “EAs híbridos” e “EAs Lamarckianos”. Uma característica particular e responsável por manter os AMs como tema atual, é que ao contrário dos métodos tradicionais de CE, os AMs estão intrinsicamente preocupados em explorar o conhecimento disponível sobre o problema em estudo (MOSCATO *et al.*, 2004). De uma maneira mais simples, um AM é aquele que apresenta o uso de uma população como possíveis soluções, mas comumente designado como característica de busca global, e também algum outro tipo de busca com informações de vizinhança, que pode ser chamada de característica de busca local.

### 1.1.1 Motivação

A tecnologia teve um grande avanço nestes últimos anos, mesmo assim os recursos computacionais para certas aplicações necessitam de otimização para que os custos envolvidos em soluções de alguns problemas não sejam elevados. Segundo SILVA (2012b), o grau de dificuldade em resolver um determinado problema com um algoritmo dedicado está estreitamente relacionado com a sua complexidade computacional, ou seja, a quantidade de recursos como tempo e memória necessária para fazê-lo. O número de elementos de entrada para a aplicação do algoritmo está diretamente relacionado com a complexidade computacional. Por outro lado, temos as dificuldades que são intrínsecas a cada tipo problema devidas suas características. Os problemas do mundo real geralmente possuem características multimodais. As funções multimodais possuem diversos ótimos locais, o que as vezes fazem com que alguns algoritmos tenham uma finalização prematura quando ficam presos a um desses ótimos locais. Existe uma área bem abrangente para pesquisa de desenvolvimento de algoritmos eficientes para problemas de otimização multimodais.

A representação de problemas sujeitos a restrições, é realizada por um conjunto de funções que satisfazem um conjunto de restrições. Segundo SILVA (2012b), as restrições são importantes em problemas de projeto de engenharia, uma vez que normalmente são impostas na declaração do problema e que às vezes são muito difíceis de satisfazer, o que pode tornar a busca difícil e ineficiente.

A hibridização de algoritmos evolucionários com heurísticas de busca local, que dão origem a um AM, tem sido muito aplicada para solucionar diversos problemas (SHAHOOKAR *et al.*, 1994; MATSUMURA *et al.*, 2000; SUN *et al.*, 2009; NAITALI e GIRI, 2010; ALI e AWAD, 2014; KATSIKIANNIS e STAVRAKAKIS, 2014). O uso do algoritmo cultural também vem sendo divulgado e usado em vários trabalhos científicos (SUN *et al.*, 2009; ZHANG, 2011; ALI *et al.*, 2014; JIA e HU, 2014). Segundo NORMAN e MOSCATO (1991), o teorema *no-free-lunch* (WOLPERT e MACREADY, 1997), deixou claro que um algoritmo de busca funciona estritamente de acordo com a quantidade e qualidade do conhecimento do problema que eles incorporam, destacando assim um dos “motivos vivo” do uso dos AMs.

Com base no exposto acima, nota-se que assuntos relacionados a solução de problemas de otimização no mundo real é muito vasto. Além de que, o uso de ferramentas com algoritmos evolucionários pode proporcionar soluções adequadas as características particulares de cada problema e a hibridização com as heurísticas de busca local podem tornar estas soluções mais eficientes. O uso do AC que realiza a exploração no domínio do problema e atualiza as suas bases de conhecimentos no decorrer desta atividade, fornece para a busca local uma região de soluções promissoras para a intensificação da busca. Outro fato importante é apontado por (BROWNLEE, 2011), a informação cultural é compartilhada entre os indivíduos, espalhando-se através da população como memes em relação à sua adequação ou aptidão que os memes transmite aos indivíduos. Logo, o AM criado com base no AC e uma busca local pode fornecer excelentes resultados em sua aplicação.

### 1.1.2 Objetivos da Tese

Analisar comportamento dos Algoritmos Culturais, com populações evoluídas pelo Algoritmo Genético quando são utilizadas as heurísticas de busca local: *Tabu Search*, *Beam Search*, *Hill Climbing* e *Simulated Annealing*. Esse objetivo geral será alcançado por meio das seguintes tarefas:

- a) Desenvolver um algoritmo memético pela hibridização do algoritmo cultural com as heurísticas de busca local: *Tabu Search*, *Beam Search*, *Hill Climbing* e *Simulated Annealing*, sendo selecionadas uma por vez.

- b) Avaliar os algoritmos e o software desenvolvido através de diversos testes padrões e ensaios (*benchmark*).
- c) Avaliar o comportamento do algoritmo memético desenvolvido na solução de problemas de engenharia.

## 1.2 O Estado da Arte

Diversos trabalhos são realizados com o objetivo de mostrar novas técnicas de otimização que apresentem melhoras no desempenho, na qualidade das soluções e no custo computacional. No desenvolvimento desta tese, foram consultados diversos artigos relacionados ao uso de algoritmos evolutivos e algoritmos meméticos para a solução de problemas de engenharia com restrições e também a análise de desempenho desses algoritmos em funções benchmark da literatura. Na área de algoritmos evolucionários que tratam de busca tabu, busca em feixe e/ou algoritmo cultural temos (ALI e AWAD, 2014; WU *et al.*, 2015; BENNELL *et al.*, 2018; GUAN *et al.*, 2018). Algoritmos meméticos e/ou busca local (NERI e KHAN, 2014; WANG *et al.*, 2014; LIN, G. *et al.*, 2016).

Desde sua apresentação ao mundo científico, os AMs, tem sua aplicação de forma contínua em diversas áreas do conhecimento. Uma das áreas mais comuns de aplicação dos AMs, é a dos tradicionais problemas NP-difícil de otimização, onde é notável a história de sucesso confirmada pela consulta em diversos trabalhos (NALEPA e BLOCHO, 2016; SHANG *et al.*, 2016; ZHANG *et al.*, 2017; KÓCZY *et al.*, 2018; ZHOU *et al.*, 2018; FENG *et al.*, 2019). Alguns trabalhos abordaram o uso de AMs em problemas de otimização aplicados na área de redes e telecomunicações (KIM *et al.*, 2007; SALCEDO-SANZ e YAO, 2008; SEGREDO *et al.*, 2011; ZHU *et al.*, 2012; CHEN *et al.*, 2014; MIRSALEH e MEYBODI, 2018). Segundo MOSCATO *et al.* (2004), os problemas de programação são um dos domínios de otimização mais importantes, devido à sua importância no Planejamento da Produção, apesar de poderem ser incluídos na classe NP-difícil. Os AMs aplicados em problemas de programação são largamente utilizados e a cada nova aplicação são obtidos resultados melhores (BURKE e SMITH, 1999; XHAFA *et al.*, 2008; LIU *et al.*, 2014; SHEN *et al.*, 2018; DECERLE *et al.*, 2019; MIGUEL *et al.*, 2019; WU e CHE, 2019). Existem aplicações de AMs, nas engenharias, na eletrônica, no eletromagnetismo, medicina, aprendizado de máquina, em robótica e diversas outras áreas. Os trabalhos aplicados nestas áreas com uso de AMs, geralmente se transformam em aplicações práticas (GRÉWAL *et al.*, 2006; TIRRONEN *et al.*, 2008; POMBO *et al.*, 2015; KYRIACOU *et al.*, 2017; SAN-JOSÉ-REVUELTA, 2018; WELEKAR e THAKUR, 2019).

Segundo BROWNLEE (2011), o Algoritmo Cultural é uma extensão do campo da Computação Evolutiva e pode ser considerado um Algoritmo Meta-Evolutivo. Mas, pertence amplamente ao campo da Inteligência Computacional e Metaheurísticas. Está relacionado a outras extensões de alta ordem da Computação Evolucionária, como o Algoritmo Memético. Alguns autores têm utilizado a característica do AC de exploração total do domínio definido e a vantagem de atualizar seus indivíduos através das suas bases de conhecimento, em conjunto com algum tipo de busca local para intensificar a localização do objetivo global na vizinhança dos valores encontrado pelo AC. O AM resultante da hibridização do AC com uma busca local é usado por NGUYEN e YAO (2006), este aproveita o conhecimento da exploração realizada e mantida pelo AC para guiar a busca local com o intuito de melhorar o resultado obtido pelo AC. Outros autores também têm criado AMs com o uso do AC e busca local. Porém, seguem diversas maneiras para aplicação, criando nichos, dividindo as populações e na maioria das vezes só se utilizam de uma fonte de conhecimento, ou então, usam estas fontes na sua forma clássica (DIGALAKIS e MARGARITIS, 2002; KOBTI, 2013). Porém, a quantidade de artigos que abordam o AC com busca local é muito pequena e geralmente não são aplicados em problemas com restrições e na pesquisa bibliográfica realizada não foram encontrados trabalhos desta natureza resolvendo problemas reais de energia na forma de otimização multiobjetivo. A tabela 1.1, apresenta de alguns trabalhos publicados nas áreas de aplicação desta tese que vão desde, mostrando publicações atuais e alguma anteriores que dão suporte a contínua pesquisa.

**Tabela 1.1:** Processo físico de recozimento versus o algoritmo de recozimento simulado.

Algoritmos evolucionários que tratam de <b>busca tabu, busca em feixe e/ou algoritmo cultural.</b>	(ALI e AWAD, 2014; WU et al., 2015; BENNELL et al., 2018; GUAN et al., 2018)
<b>Algoritmos meméticos</b> e/ou busca local.	(NERI e KHAN, 2014; WANG et al., 2014; LIN, G. et al., 2016)
Aplicação de <b>Algoritmos Meméticos</b> em tradicionais <b>problemas de otimização NP-difícil.</b>	(NALEPA e BLOCHO, 2016; SHANG et al., 2016; ZHANG et al., 2017; KÓCZY et al., 2018; ZHOU et al., 2018; FENG et al., 2019).
Uso de <b>Algoritmos Meméticos em problemas de otimização</b> aplicados na área de redes e telecomunicações.	(KIM et al., 2007; SALCEDO-SANZ e YAO, 2008; SEGREDO et al., 2011; ZHU et al., 2012; CHEN et al., 2014; MIRSALEH e MEYBODI, 2018). Segundo MOSCATO et al. (2004)

Os <b>Algoritmos Meméticos</b> aplicados em problemas de programação são largamente utilizados e a cada nova aplicação são obtidos resultados melhores	(BURKE e SMITH, 1999; XHAFA et al., 2008; LIU et al., 2014; SHEN et al., 2018; DECERLE et al., 2019; MIGUEL et al., 2019; WU e CHE, 2019)
Aplicações de <b>Algoritmos Meméticos nas engenharias</b> , na eletrônica, medicina, no eletromagnetismo, aprendizado de máquina, em robótica e diversas outras áreas	(GRÉWAL et al., 2006; TIRRONEN et al., 2008; POMBO et al., 2015; KYRIACOU et al., 2017; SAN-JOSÉ-REVUELTA, 2018; WELEKAR e THAKUR, 2019)

A seguir são apresentados dez artigos selecionados que estão na faixa de publicação 2014 a 2018. Os mesmos são atuais, usam algoritmos cultural e/ou heurísticas de busca local, são híbridos e confirmam que o desempenho não é aferido por meio de testes estatísticos eficientes e que não são aplicados em problemas multiobjetivo de sistemas de potência reais.

### 1.2.1 *A two-phase tabu-evolutionary algorithm for the 0-1 multidimensional knapsack problem*

No trabalho de LAI *et al.* (2018), é apresentado o algoritmo evolucionário tabu de duas fases (TPTEA), para resolver o problema da mochila multidimensional 0–1 que é um problema de otimização combinatória NP-difícil. O (TPTEA) proposto depende particularmente de dois procedimentos de busca tabu baseados em solução para explorar diferentes espaços de busca.

Segundo o autor esta é a primeira vez que a busca tabu baseada em soluções é utilizada para resolver o MKT e os dois procedimentos de busca dedicados para explorar diferentes espaços de busca. Estes são integrados em um framework evolucionário de base populacional para que seja capaz de garantir uma efetiva intensificação e diversificação dentro do espaço de busca.

Para que o algoritmo tenha um melhor desempenho, a população inicial é constituída por um procedimento randômico que gera várias soluções viáveis e após diversas iterações e quando todos os itens forem verificados quanto ao atendimento das restrições de mochila, o procedimento de inicialização é interrompido.

Foi mostrada a competitividade do algoritmo proposto, apresentando resultados computacionais em 281 instâncias de benchmark comumente usadas na literatura. Em particular, em uma comparação computacional com os melhores algoritmos da literatura em múltiplos conjuntos de dados, mostrou-se que o método em média corresponde mais do que o dobro do número de soluções mais conhecidas para os problemas mais difíceis do que qualquer outro método e soluções (novos limites inferiores) para 4 instâncias difíceis.

### 1.2.2 *The Bounded Beam Search algorithm for the Block Relocation Problem*

O artigo de BACCI *et al.* (2018), trabalha na solução do problema restrito de realocação de blocos, que possui aplicações práticas relevantes na logística de contêineres.

Uma aplicação real do problema de realocação de blocos (*Block Relocation Problem* – BRP) surge na logística de contêineres em um terminal. Um terminal de contêineres é uma área onde é realizado o transbordo entre diferentes veículos de transporte, como navios de carga, trens, caminhões de contêineres e onde eles são empilhados devido ao espaço de armazenamento limitado. A área de armazenamento (pátio) é geralmente dividida em grupos de pilhas de contêineres, chamadas de baias, e os contêineres são movidos por guindastes de pátio.

O BRP consiste em decidir onde realocar cada bloco que é movido por uma operação de remodelação, para minimizar o número total de remodelações necessárias para recuperar todos os blocos de acordo com a ordem de recuperação  $(1, \dots, n)$ . O mínimo é o limite inferior que são os números de blocos de bloqueio de  $S$ , ou seja, aqueles localizados em qualquer slot acima de um bloco com maior prioridade de recuperação.

O autor apresentou um novo limite inferior e uma abordagem heurística para o problema. Usou esta abordagem dentro de um algoritmo de busca de feixe limitado para resolver o problema de realocação de blocos com o intuito de mostrar que a abordagem heurística considerada supera os outros algoritmos existentes na maioria das instâncias da literatura. Introduziu novas grandes instâncias do BRP para testar as abordagens em dimensões de tamanho real.

### 1.2.3 *Developing a Dynamic Neighborhood Structure for an Adaptive Hybrid Simulated Annealing – Tabu Search Algorithm to Solve the Symmetrical Traveling Salesman Problem*

O trabalho de LIN, Y. *et al.* (2016), faz aplicação de um algoritmo híbrido meta-heurístico adaptativo que combina algoritmos de busca simulada e busca tabu com uma estrutura de vizinhança dinâmica para resolver o Problema do Caixeiro Viajante (TSP). Foram consideradas as características do algoritmo híbrido, desenvolvendo uma estrutura de vizinhança dinâmica para o algoritmo híbrido afim de melhorar a eficiência de busca, reduzindo a aleatoriedade da vizinhança convencional de 2-opt. Os autores desenvolveram uma mutação dirigida por um círculo para alcançar a estrutura de vizinhança dinâmica. Além de propor parâmetros adaptativos que podem ser ajustados automaticamente pelos algoritmos baseados

em exemplos específicos do contexto. Isso impede a necessidade de reajustar freqüentemente os parâmetros do algoritmo.

Os autores empregaram benchmarks obtidos a partir do TSPLIB (uma biblioteca de exemplos de instâncias para o TSP) para testar o algoritmo proposto, e descobriram que o algoritmo proposto pode obter soluções satisfatórias dentro de um período de tempo razoável. Os resultados experimentais demonstram que o algoritmo híbrido proposto pode superar as desvantagens dos métodos tradicionais de recozimento simulado e busca tabu. Os resultados também mostram que a estrutura de vizinhança dinâmica é mais eficiente e precisa do que a clássica 2-opt. Além disso, parâmetros adaptativos são apropriados para quase todos os exemplos numéricos testados neste artigo. Finalmente, os resultados experimentais são comparados com os de outros algoritmos, para demonstrar a maior precisão e eficiência do algoritmo proposto.

#### *1.2.4 A novel class of niche hybrid Cultural Algorithms for continuous engineering optimization*

O artigo de ALI e AWAD (2014), apresenta três algoritmos desenvolvidos com base no método evolutivo do algoritmo cultural. A proposta é fornecer uma nova classe de nicho de algoritmo cultural híbrido. O primeiro algoritmo trata-se de um nicho de algoritmos culturais (NCA), que contém um *framework* que mantém múltiplos grupos dentro de agentes de população a fim de localizar múltiplos ótimos locais. O segundo algoritmo é uma hibridização do nicho de algoritmos culturais com a técnica de busca local denominada de busca tabu (H-NCA). Este esquema de hibridização possibilita o algoritmo transpor ótimos locais e melhorar o desempenho. O terceiro algoritmo é o híbrido (H-NCA) melhorado, ou seja, (IH-NCA). Este faz o chaveamento entre duas estratégias de seleção a roleta e o torneio estocástico. Ele aumenta a taxa de convergência e a precisão, pois escapa da convergência prematura e da estagnação.

Para o NCA, o Algoritmo Cultural gera a população chamada original, depois é aplicado o mecanismo de *niching-clearing* (PÉTROWSKI, 1996; SACCO *et al.*, 2004), onde são formadas subpopulações dentro da população original. Cada nova população é classificada em ordem decrescente para que dentro de um determinado raio de atuação sejam encontrados os indivíduos “*top*” que serão selecionados como os vencedores do nicho, os demais indivíduos serão desprezados. Desta forma cada nicho terá seu vencedor de onde será retirado o vencedor global.

O H-NCA, realiza a hibridização do NCA com a busca tabu que aproveita a solução encontrada pelo NCA (*NCA\_best*) como semente a cada iteração, ou seja, solução inicial. Em



seguida uma sequência de movimentos é realizada para encontrar todas as possíveis soluções vizinhas associadas ao *NCA\_best*. A lista tabu armazena as últimas soluções visitadas (soluções proibidas). Um movimento do tabu que começa a partir da solução atual (*NCA\_best*) com uma solução encontrada na lista Tabu não pode ser um membro de regiões vizinhas de *NCA\_best*. O melhor membro é então escolhido como tendo a melhor aptidão para ser comparado com *NCA\_best* do estágio anterior do algoritmo. O objetivo desta hibridização foi encontrar uma solução melhor do que a encontrada por NCA e ajustá-lo até que o critério de parada está satisfeito. O IH-NCA, herda as características do H-NCA e melhora seu desempenho pelo uso de uma estratégia de seleção adaptativa que usa dois tipos de estratégias de seleção para melhorar o trabalho da função de aceitação na AC.

A função aceitação modificada usa estratégia de roleta e uma seleção roleta modificada usando a seleção torneio estocástica.

Este também apresenta de forma simplificada o conceito básico de algoritmo cultural e busca tabu. Fornece informações sobre diversos algoritmos utilizados para *benchmark*. Os métodos de comparações e avaliações obtidos, podem ser usados em diversos trabalhos que buscam apresentar melhorias para algoritmos de otimização.

### 1.2.5 *Opposition-based Memetic Search for the Maximum Diversity Problem*

Uma proposta para resolver o problema de diversidade máxima (MDP – *maximum diversity problem*), que é um desafio computacional, é apresentada por ZHOU *et al.* (2017). A proposta é um algoritmo memético baseado em oposição (OBMA – *Opposition-based memetic algorithm*), que integra o conceito de aprendizagem baseada em oposição (OBL – *opposition-based learning*) em uma estrutura memética de busca bem conhecida. O OBMA explora as soluções candidatas e suas soluções opostas durante seus processos de inicialização e evolução. Combinado com um poderoso procedimento de otimização local e uma estratégia de atualização de *pool* de qualidade e distância baseada em classificação, o OBMA estabelece um equilíbrio adequado entre exploração e a intensificação de seu processo de busca.

No algoritmo proposto pelos autores, é usado um procedimento de busca baseado em oposição de dupla trajetória (ODTS – *opposition-based double trajectory search*), onde simultaneamente é realizada a busca em torno de uma solução descendente e da solução oposta. A otimização local usada foi uma melhoria no procedimento de busca tabu em vizinhança com restrição paramétrica.

O procedimento melhorado de busca tabu apresentado pelos autores se distingue dos demais por sua vizinhança com restrição paramétrica que permite ao processo de busca explorar soluções candidatas mais promissoras.

### **1.2.6** *Path-relinking Tabu search for the multi-objective flexible job shop scheduling problem*

No artigo de JIA e HU (2014), o problema de agendamento flexível de tarefas é resolvido usando um novo algoritmo de caminho religado (PR – *path-relinking*) que se baseia no algoritmo de busca tabu com rastreamento de *back-jump*. O algoritmo proposto possui três características distintas. A primeira é de explorar o espaço de busca de forma inteligente através da heurística PR. A segunda é o mecanismo efetivo de dimensão orientada (IS – *Intensification search*). A terceira é o algoritmo TSAB (*Tabu search algorithm with back-jump*) estendido, para resolver problemas de otimização multiobjetivo.

A solução de roteamento é identificada pela pesquisa de vizinhança específica do problema e, em seguida, é refinada ainda mais pelo TSAB para uma decisão de sequenciamento. A solução resultante é usada para manter a memória de médio prazo, onde as melhores soluções são armazenadas. Uma heurística PR é projetada para gerar diversas soluções nas áreas mais promissoras. Para obter uma versão melhorada do algoritmo foi incorporado a pesquisa IS orientada localizadas perto de soluções extremas.

Os autores finalizam informando que os resultados comparativos mostram que os algoritmos propostos são competitivos em termos de desempenho computacional e qualidade da solução.

### **1.2.7** *Balancing search direction in cultural algorithm for enhanced global numerical optimization*

O artigo de ALI *et al.* (2014), utiliza uma versão modificada do algoritmo cultural (CA) que usa quatro fontes de conhecimento (Normativo, de domínio, topográfico e o situacional) no intuito de incorporar informação obtida da função objetivo bem como violação de restrições dentro da estrutura de conhecimento no espaço de crença. Em um primeiro momento é realizada uma busca com base nas quatro fontes de conhecimento e com uma exploração e aproveitamento balanceado para guiar o processo de busca. De uma forma geral também é explorada as múltiplas trajetórias de busca.

No início da pesquisa, todas as fontes de conhecimento têm a mesma probabilidade de serem escolhidas. No entanto, em gerações posteriores, a probabilidade de escolher uma fonte

de conhecimento depende se ele é um explorador (normativo e topográfico) ou descobridor (situacional e domínio). A fase posterior da pesquisa vai determinar onde os indivíduos vão começar a se mover, dependendo da direção de busca.

Em cada geração, a qualidade das soluções criadas na geração anterior irá determinar a porcentagem de indivíduos da população partilhada ou número de seguidores para cada uma destas fontes de conhecimento. O indivíduo é selecionado dos melhores e médios desempenhos para geração atual. Em seguida, uma série de soluções vizinhas é gerada através da pesquisa local (memética) para encontrar uma melhor solução. Os valores de fitness das soluções recém-geradas são calculados e comparados com o atual melhor. Todo o processo é repetido até que a condição de parada seja satisfeita.

Após as simulações foram comparados os resultados com outros algoritmos usando para solucionar problemas de otimização numérica e o resultado demonstrou um alto desempenho e menor custo computacional.

### *1.2.8 Energy and Labor Aware Production Scheduling for Industrial Demand Response Using Adaptive Multi-Objective Memetic Algorithm*

No artigo de GONG *et al.* (2018), é proposto um Algoritmo memético multiobjetivo adaptativo (AMOMA - *Adaptive Multiobjective Memetic Algorithm*) baseado no NSGA-II para otimizar um modelo integrado de programação de produção de energia e mão-de-obra com preços de eletricidade em tempo real. As contribuições deste artigo são triplas.

(1) Comparado aos modelos existentes de programação de produção de energia, o modelo proposto considera adicionalmente o tipo e a quantidade de trabalho, o turno de trabalho, bem como os períodos de produção proibidos. Isso torna a mudança de carga industrial mais realista.

(2) O AMOMA proposto integra-se sinergicamente nas pesquisas tabu (TS – *Tabu Search*) orientadas para convergência e diversidade do NSGA-II, respectivamente. Além disso, coordena de forma adaptativa a exploração e a intensificação durante uma pesquisa.

(3) Um estudo de caso em uma máquina de Moldagem de Extrusão por sopro (EBM - *Extrusion Blow Molding*) foi realizado. Usando dados empíricos e extensos benchmarks, os AMOMA propostos comprovadamente alcançam uma rápida aproximação da frente de Pareto (PF – *Pareto Front*) enquanto preservam a diversidade para este problema de otimização multiobjetivo (MOP - *multiobjective optimization problem*) altamente restrito.

### 1.2.9 *A Tabu search based hybrid evolutionary algorithm for the max-cut problem*

O artigo de WU *et al.* (2015), apresenta um algoritmo evolucionário híbrido baseado em busca tabu (TSHEA - *Tabu Search Based Hybrid Evolutionary Algorithm*) para resolver o problema corte máximo (*max-cut*).

Nesta proposta o componente de pesquisa tabu alterna entre duas fases de pesquisa, uma das quais é baseada em um movimento de uma volta e a outra depende de movimentos de troca restritos. O operador de combinação de soluções herda componentes de soluções comuns de soluções pai e completa a parte restante usando uma avaliação de qualidade e distância, que pode ser considerada como uma variante inovadora do tradicional *crossover* uniforme.

As comparações estatísticas indicam que o algoritmo TSHEA proposto é significativamente melhor do que vários algoritmos atuais de última geração. Além disso, foram examinados vários componentes importantes do TSHEA, incluindo o operador de combinação de solução, a vizinhança combinada e as configurações dos parâmetros. A análise revela seu mérito para o alto desempenho do algoritmo TSHEA.

Para esta proposta os três principais aspectos são:

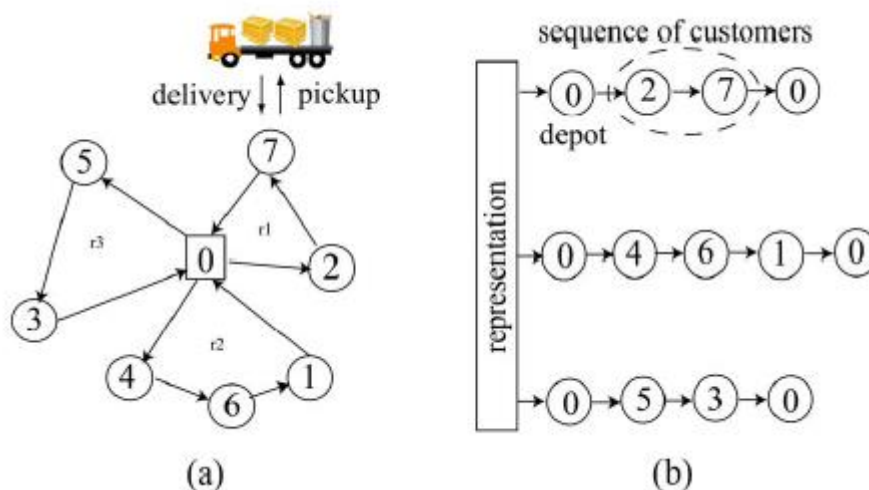
- (1) TSHEA usa a combinação de vizinhos no seu procedimento de busca tabu, na qual aumenta o aproveitamento da vizinhança;
- (2) TSHEA usa um operador de combinação de solução similar ao tradicional operador *crossover* pela referência de duas soluções mães;
- (3) TSHEA é avaliado em muitos casos benchmark e é capaz de encontrar melhor desempenho que o algoritmo memético (MA - *Memetic Algorithm*).

### 1.2.10 *Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, Instances, and Algorithm*

No artigo de WANG *et al.* (2016), são mostrados dois algoritmos com o propósito de resolver uma variação dos problemas de roteamento de veículos (VRP - *Vehicle Routing Problem*), denominado de VRP com simultâneas entrega, coletas e janela de tempo (VRPSDPTW – *VRP with simultaneous delivery and pickup and time windows*), que é um importante problema na rede logística de otimização da cadeia de suprimentos em malha fechada. Este artigo usou cinco objetivos baseados em dados do mundo real, pela característica multiobjetivo do problema, este ficou com a denominação de Problema multiobjetivo de roteamento de veículo com simultâneas entregas, coletas e janela de tempo (MO-VRPSDPTW – *Multiobjective - VRP with simultaneous delivery and pickup and time windows*). Os dois algoritmos desenvolvidos para solucionar o MO-VRPSDPTW, são o multiobjetivo busca local

(MOLS - *Multiobjective Local Search*) e o multiobjetivo algoritmo memético (MOMA – *Multiobjective Memetic Algorithm*), estes foram desenvolvidos, implementados e comparados. O algoritmo MOLS utiliza três operadores de vizinhança: N1, N2 e N3. Onde: N1 remove aleatoriamente um cliente de uma rota selecionada; N2, primeiramente remove aleatoriamente um número de clientes de uma rota selecionada; e N3, troca a sequência de clientes entre duas rotas, mantendo a orientação da sequência. As alterações na estrutura de vizinhança e os elementos estocásticos do MOLS permite uma eficaz diversificação para escapar de mínimos locais. O algoritmo MOMA inclui decomposição, operador *crossover* e busca local. A decomposição é realizada para decompor o problema em subproblemas usando a abordagem de soma ponderada. O crossover gera novas soluções escolhendo aleatoriamente um número de rotas dos primeiros pais e copiados para os filhos. Essa geração é melhorada pela busca local. As principais contribuições foram a introdução de uma variante de cinco objetivos de VRPSDPTW; introdução de um conjunto de instâncias realísticas de *benchmark*; desenvolvimento, implementação e teste de dois algoritmos para resolver o MO-VRPSDPTW; e a realização de extensivo experimento para avaliar os dois algoritmos propostos. A Figura 1.2 mostra a imagem de um exemplo de solução. A inicialização da população é igual para os dois algoritmos propostos. Porém, no algoritmo MOMA é usado um *framework* evolucionário MOEA/D que inclui a decomposição da população em subpopulações e utiliza o operador de crossover para depois aplicar a busca local.

**Figura 1.1:** (a) Solução e (b) Representação.



**Fonte:** (WANG *et al.*, 2016).

Nos dez subitens anteriores foram apresentados alguns trabalhos que direta ou indiretamente têm relação com a proposta apresentada neste trabalho. Ou seja, pelo uso de

algoritmos híbridos ou isolados, resolvendo problemas de otimização e/ou aplicação em problemas reais de diversas áreas.

### 1.3 *Justificativa*

Dentre os artigos consultados não foram encontrados nenhum que esteja abordando a proposta da tese que trata do desenvolvimento de um algoritmo memético pela hibridização do algoritmo cultural com de diversas heurísticas de busca local (Busca Tabu – *Tabu Search*, Busca em Feixe – *Beam Search*, Recozimento Simulado – *Simulated Annealing* e Escalando a Colina – *Hill Climbing*) para otimização de problemas de variáveis reais. Desta forma, justifica-se a realização deste trabalho, que com esta avaliação será possível verificar o comportamento das classes desenvolvidas na solução de funções multimodais e em problemas de engenharia de uma forma geral. Fornecendo uma nova alternativa para solução de problemas de otimização.

### 1.4 *Estrutura do Trabalho*

O trabalho está dividido em seis capítulos. No Capítulo 1 foi realizada uma introdução ao assunto com as considerações iniciais e o estado da arte. No capítulo 2 é apresentada a fundamentação teórica onde tem-se: algoritmos evolutivos, os algoritmos culturais (ACs), um breve resumo do algoritmo genético (AG), técnicas de busca, abordagem memética de uma maneira geral, reforçando o uso do algoritmo cultural e as buscas locais a partir das referências já publicadas na literatura e técnicas de medidas de desempenho de algoritmos. No capítulo 3 será apresentada a metodologia do trabalho. No capítulo 4 é apresentado o algoritmo memético cultural que é o algoritmo desenvolvido nesta tese. O capítulo 5 apresenta os resultados dos experimentos realizados (funções *benchmark* e os problemas de engenharia com restrições), no capítulo 6 tem-se as conclusões e trabalhos futuros, e por último mostra-se a bibliografia utilizada nesta tese.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Algoritmos Evolutivo

Na busca de encontrar soluções de problemas computacionais complexos, a Computação Evolutiva (CE) procura desenvolver conceitos e algoritmos baseados em sistemas naturais, mas basicamente em mecanismos de adaptação dos seres vivos, como a adaptação das espécies ao ambiente natural, por meio do processo evolutivo, ou a busca de alimento efetuada de maneira coordenada por colônias de formigas ou bando de pássaros (GASPAR-CUNHA *et al.*, 2012). Esta área de pesquisa teve um grande desenvolvimento nos 40 últimos anos do século XX, isto pode ser visto pelos trabalhos de MOSCATO (1989); REYNOLDS e ZANONI (1992); ZHANG e KIM (2000); LIAO e TSAO (2002); REYNOLDS e ALI (2008); GASPAR-CUNHA *et al.* (2012) e WU *et al.* (2015). Vários motivos ajudaram neste rápido desenvolvimento, podemos citar que a capacidade de encontrar soluções adequadas para problemas complexos através de algoritmos desenvolvidos com esta abordagem foi um deles. Estes algoritmos que utilizam os princípios básicos da teoria da evolução e da genética são chamados de Algoritmos Evolutivos (AE), estes possuem métodos simples e podem ser modelados por poucas linhas de códigos. O AE possui adaptação fácil para diversas áreas, tais como: ciências naturais, biologia, economia, biologia, engenharia, ciência da computação, etc. Com o avanço e a disponibilidade de computadores o AE teve seu desenvolvimento intensificado. As pesquisas começaram a modelar a evolução natural dos seres vivos em sistemas computacionais. O mecanismo heurístico de adaptação iria tratar simultaneamente um conjunto de diferentes vetores de variáveis de decisão (esse conjunto seria o análogo computacional de uma “população” de seres vivos), (GASPAR-CUNHA *et al.*, 2012).

Em função deste mecanismo de adaptação, foram desenvolvidas de forma independentes quatro abordagens de AE: as Estratégias Evolutivas (EE) (DE JONG, 2006), a

programação evolutiva (PE) (DE JONG, 2006), a programação genética (PG) (GASPAR-CUNHA *et al.*, 2012) e o algoritmo genético (AG) (DE JONG, 2006). Com o avanço das pesquisas verificou-se que heurísticas dotadas de mecanismos distintos de funcionamento podem ser mais adequadas para problemas com determinadas estruturas, e outras heurísticas podem funcionar melhor em outras classes de problemas (GASPAR-CUNHA *et al.*, 2012). Isto conduziu as pesquisas para o desenvolvimento de novas heurísticas que se basearam em outros processos da natureza diferentes do da evolução da espécie. Dentre essas se podem citar o sistema de colônia de formigas (GASPAR-CUNHA *et al.*, 2012) e o sistema imunológico artificial (FIGUEREDO *et al.*, 2013). Devido essa nova abordagem de aumento no conhecimento dos mecanismos que fundamentam os algoritmos da computação evolutiva, nota-se que os novos AE estão se afastando da estrita inspiração biológica. Os novos AEs tendem a aprofundar a tendência de incorporação de operações e mecanismos que não sejam bio-inspirados, mas sim inspirados em argumentos matemáticos ou computacionais (GASPAR-CUNHA *et al.*, 2012). Os algoritmos evolutivos mais utilizados são: os algoritmos genéticos, estratégias evolutivas, programação genética, otimização por enxame de partículas, otimização por colônias de formigas, sistema imunológico artificial, evolução diferencial e algoritmos meméticos (SILVA, 2012b).

## 2.2 Algoritmo Genético (AG)

### 2.2.1 Considerações Iniciais

Os algoritmos genéticos (AGs) desenvolvidos por John Holland, são métodos de pesquisa probabilísticos inspirados nos princípios da seleção natural e da genética (GASPAR-CUNHA *et al.*, 2012). A pesquisa buscava fazer um estudo dos mecanismos de adaptação existente na natureza e modelar computacionalmente os princípios básicos identificados nos sistemas naturais. Holland estudou a evolução natural considerando esta como um processo robusto, simples e poderoso que poderia ser adaptado para obtenção de soluções computacionais eficientes para problemas de otimização (GABRIEL e DELBEM, 2008). Goldberg foi capaz de resolver um problema difícil, envolvendo o controle de transmissão em gasoduto em sua dissertação e resumiu o trabalho de Holland em um livro (SILVA, 2012b). Os AGs fazem parte da classe de algoritmos evolucionários baseados na teoria de Darwin pela qual pressupõe que indivíduos com boas características genéticas têm maiores chances de sobreviver e produzir indivíduos mais aptos em uma dada população (SANTOS, 2007). A abordagem genética fornece a capacidade de identificar e explorar aspectos do ambiente, ou seja, possui a



capacidade de busca e movimentação no espaço de solução. Geralmente, o AG realiza uma busca global em todo espaço procurando por soluções ótimas ou as melhores soluções possíveis. Esta ferramenta tem se mostrada muito eficiente para problemas de busca e otimização de diversos tipos e aplicações. Todas essas características fazem do AG uma ferramenta de otimização realmente eficaz no processo de otimização (SIVANANDAM e DEEPA, 2007). O trabalho de DE JONG (1975) demonstrou a utilidade do AG para a otimização de funções e fez o primeiro esforço concentrado para encontrar parâmetros otimizados para AGs (SONKAR *et al.*, 2012).

### 2.2.2 Características dos Algoritmos Genéticos (AGs)

Os algoritmos genéticos diferem dos métodos tradicionais de busca e otimização, principalmente em quatro aspectos (GOLBERG, 1989):

- Trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros;
- Trabalham com uma população e não com um único ponto;
- Utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar. Por isso não há necessidade de derivadas ou qualquer outro tipo de informação adicional, o que facilita sua aplicação em situações onde os dados são discretos e não possuem derivadas (LINDEN, 2008);
- Utilizam regras de transição probabilísticas e não determinísticas.

Por possuírem regras de transição probabilísticas, estes podem encontrar soluções diferentes todas as vezes que forem executados, mesmo que os parâmetros sejam fixados e utilizem a mesma população inicial.

Os AGs são caracterizados como heurísticas de busca no espaço de soluções e diferentemente dos esquemas enumerativos, não realizam buscas em todos os pontos de soluções possíveis, mas apenas em subconjuntos desses pontos (SILVA, 2012b). Estes empregam uma estratégia de busca paralela e estruturada, embora aleatória, direcionada à busca de pontos de “alta aptidão”, ou seja, pontos nos quais a função a ser minimizada ou maximizada tem valores relativamente baixos ou altos (respectivamente) (SANTOS, 2007). Dessa forma, é correto afirmar que os AGs não podem ser chamados de buscas aleatórias não direcionadas (LINDEN, 2008). PHAM e KARABOGA (2012) também conclui que o AG é uma técnica de busca aleatória direcionada que pode encontrar uma solução ótima global em um complexo espaço de busca multidimensional.

De uma forma básica, os AGs operam formando inicialmente um conjunto de indivíduos (geração inicial) que são possíveis soluções do problema. No decorrer do processo a população é avaliada, ou seja, é fornecido para cada indivíduo um valor que corresponde à sua aptidão, este valor reflete a habilidade de adaptação em determinado ambiente. Após o recebimento desta nota, um percentual de indivíduos mais adaptados é mantido, enquanto os demais são descartados por um processo de seleção. A avaliação do indivíduo influencia na aplicação dos operadores genéticos, tendo-se em vista a sobrevivência do indivíduo mais apto (SIVANANDAM e DEEPA, 2007). Os que foram mantidos podem sofrer alterações em suas características fundamentais através dos operadores de mutação e cruzamento (recombinação) genético, gerando descendentes para a próxima geração. Os indivíduos que possuem maiores chances de reprodução são os com maior adaptação relativa. Uma política de elitismo pode ser aplicada colocando os melhores indivíduos na próxima geração e evitando que estes desapareçam da população manipulada pelos operadores genéticos.

Durante cada geração, os princípios de seleção e reprodução são aplicados a uma porcentagem de candidatos que pode variar, dependendo da complexidade do problema e dos recursos computacionais disponíveis (SANTOS, 2007). A solução satisfatória só é encontrada após diversas iterações conforme o critério de parada definido. Embora possam parecer simplistas do ponto de vista biológico, esses algoritmos são suficientemente complexos para fornecer mecanismos poderosos e robustos de busca adaptativa (REZENDE, 2003).

### 2.2.3 Algoritmo Genético Simples

Trata-se do algoritmo genético básico que segue o mecanismo mostrado na Figura 2.1, e o fluxograma da Figura 2.2.

**Figura 2.1:** Algoritmo Básico de um AG.

```

Algoritmo Genético Simples
{
    Definindo
    {
        função desempenho;
        formação do indivíduo e tamanho da população;
        probabilidade dos operadores;
    }
    Inicializar população aleatória;
    Enquanto não alcançar critério de convergência faça
    {
        avaliar os indivíduos da população;
        executar seleção;
        executar cruzamento e mutação;
    }
}

```

**Figura 2.2:** Fluxograma de um Algoritmo Genético Básico.



O AG é útil e eficaz quando (SILVA, 2012b):

- O espaço de busca é grande, complexo ou mal compreendido;
- O conhecimento sobre o domínio é escasso ou o conhecimento especialista é difícil de codificar para um espaço de busca consideravelmente menor;
- Não existe possibilidade de análise matemática do problema;
- Os métodos de busca tradicionais falham.

Segundo CHAMBERS (1999), os cinco passos básicos para resolver um problema usando o algoritmo genético simples são:

- A codificação do cromossomo de modo a representar o problema;
- Uma população inicial de soluções;
- O cálculo da função aptidão (*fitness*);
- Método de seleção de seleções para produzir novas soluções;
- Operadores de recombinação e mutação para produzir novas soluções através das soluções já existentes.

## 2.3 Algoritmo Cultural (AC)

### 2.3.1 Considerações Iniciais

Dentro do processo evolucionário que antes era basicamente inspirado na evolução biológica tem-se os algoritmos culturais. O algoritmo cultural é uma estratégia de otimização global que simula a relação entre os indivíduos na sociedade e na cultura social em torno deles (SUN *et al.*, 2009). Este é usado para modelar a evolução da componente cultural de um sistema evolutivo computacional ao longo do tempo, uma vez que acumula experiência na resolução em um conjunto de dados na resolução de problemas (WEI *et al.*, 2013). A evolução cultural permite que as sociedades envolvam ou adaptem seu meio ambiente a taxas que excedem a evolução biológica, que se baseia apenas na herança genética (REYNOLDS, 1994).

### 2.3.2 Características culturais

As características culturais são guardadas em *memes*, que fazendo uma analogia às características genéticas estão diretamente relacionados aos *genes*. *Memes* são, portanto, a unidade de transmissão cultural ou imitação tal como um pedaço de pensamento, um fragmento de música e assim por diante (SILVA, 2012b).

No livro “O Gene Egoísta” Richard Dawkins cunhou o termo *meme* (DAWKINS, 2017):

“O meme é para a memória o análogo do gene para genética, ou seja, a sua unidade mínima. O meme também é considerado como uma unidade de informação que se multiplica de cérebro em cérebro, ou entre locais onde a informação é armazenada (como livros) e outros locais de armazenamento ou cérebros. Podem ser ideias ou partes das ideias, línguas, sons, desenhos, capacidades, valores estéticos e morais, ou qualquer outra coisa que possa ser aprendida facilmente e transmitida enquanto unidade autônoma”.

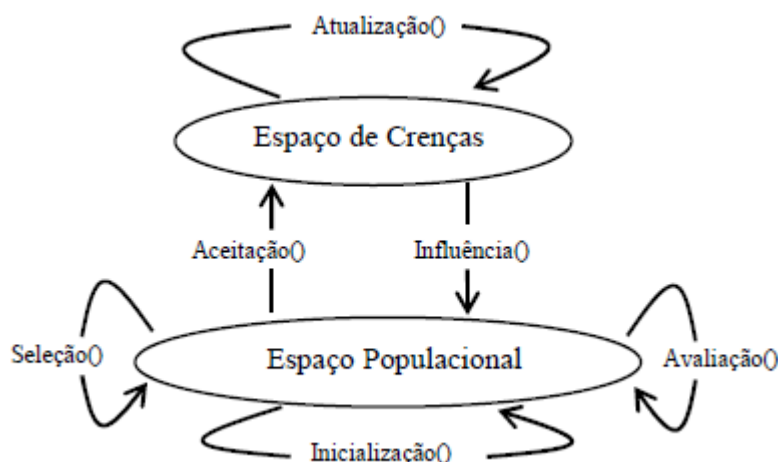
A principal vantagem da cultura é o chamado mecanismo adaptativo: a capacidade de responder ao meio de acordo com mudança de hábitos, mais rápida do que uma possível evolução biológica (SILVA, 2012b).

### 2.3.3 Funcionamento de um Algoritmo Cultural

Nas sociedades humanas, a cultura pode ser vista como um veículo para codificação, generalização e de armazenamento do conhecimento potencialmente acessível a todos membros

da sociedade (SUN *et al.*, 2009). O fluxograma que representa o funcionamento básico do AC é mostrado na Figura 2.3.

**Figura 2.3:** Fluxograma do Algoritmo Cultural.



**Fonte:** Adaptação de (REYNOLDS *et al.*, 2010).

Da Figura 2.3, pode-se notar a existência de dois componentes que são estrutura de dados e seis funções básicas que fazem com que o AC funcione dentro de suas características. Os dois componentes principais de um Algoritmo Cultural são (SILVA, 2012b):

- **Espaço Populacional:** conjunto de soluções que pode ser modelado utilizando qualquer técnica de Inteligência Computacional que faça uso de uma população de indivíduos;
- **Espaço de Crença (Mapa do grupo):** local onde ocorre o armazenamento e representação do conhecimento (experiência ou mapas individuais) adquirido ao longo do processo evolutivo. É a partir desse conhecimento armazenado que os indivíduos são guiados na direção das melhores regiões do espaço de busca.

O Espaço populacional e o espaço de crença são ligados por um mecanismo (protocolo) de comunicação composto por uma função de aceitação que é usada para coletar a experiência de indivíduos da população selecionada. A execução da função aceitação pode gerar uma modificação no espaço de crença através da função atualização. A outra função do protocolo de comunicação é a função de influência que pode fazer uso do conhecimento de soluções de problemas no espaço de crença para orientar a evolução de indivíduos no espaço populacional. O AC pode explorar tanto a microevolução quanto a macro-evolução. A microevolução diz respeito à evolução que acontece no nível populacional e a macro-evolução é a que ocorre sobre a cultura em si, ou seja, a evolução do espaço de crenças (SILVA, 2012b).

### 2.3.4 Características do Algoritmo Cultural

Segundo REYNOLDS (2002), as principais características demonstradas por um Algoritmo Cultural são:

- **Mecanismo Dual de Herança:** são herdadas características tanto no nível da população quanto no nível do espaço de crenças;
- **Evolução Guiada por Conhecimento:** a população é guiada na direção que, segundo o conhecimento armazenado no espaço de crenças, seja a melhor;
- **Suporta Hierarquização:** tanto a população quanto o espaço de conhecimento podem ser organizados de forma hierárquica, permitindo a criação de nichos e, ao mesmo tempo, uma distribuição do conhecimento adquirido;
- **Conhecimento Sobre o Domínio Separado dos Indivíduos:** todo o conhecimento adquirido é armazenado no espaço de crenças e compartilhado entre os indivíduos; assim, quando um indivíduo é eliminado da população, o conhecimento adquirido permanece;
- **Suporte a Auto Adaptação em Vários Níveis:** permite tanto a auto adaptação da população quanto do conhecimento e a forma como o conhecimento é adquirido. Ou seja, os parâmetros de controle, a representação, os operadores (tanto genéticos quanto sociais), a avaliação dos indivíduos e o protocolo de intercomunicação podem ser alterados a qualquer momento da evolução;
- **Diferentes Taxas de Evolução:** a evolução das populações e do conhecimento não precisa ocorrer na mesma taxa. Segundo REYNOLDS e ZANONI (1992), o conhecimento é evoluído a uma taxa dez vezes maior que a população;
- **Funcionamento:** é um modelo computacional que permite a modelagem de diversas formas de evolução cultural.

### 2.3.5 Espaço Populacional

Onde para cada micro passo evolutivo, ou seja, indivíduos nomeados de cultura são evoluídos ao nível evolutivo micro usando operações socialmente motivadas, tais como reprodução e aprendizagem (NAITALI e GIRI, 2010). O espaço populacional contém a população a ser evoluída e os mecanismos para sua avaliação, modificação e reprodução (COELHO e ALOTTO, 2009). O espaço populacional pode usar qualquer modelo de população evolutiva, como algoritmo genético, programação genética e programação evolucionária (JIN e REYNOLDS, 2000).

### 2.3.6 Espaço de Crenças

Todo conhecimento adquirido durante o processo evolutivo é guardado no espaço de crenças. Dentro do espaço de crenças, um conhecimento adquirido não é perdido mesmo que o indivíduo que adquiriu tal conhecimento se perca no processo de evolução. Em um algoritmo cultural a informação adquirida por um indivíduo da população pode ser compartilhada com a população inteira (COELHO e ALOTTO, 2009). As fontes de conhecimento cultural são responsáveis por coletar informações no espaço de busca e no domínio do problema para orientar os indivíduos no cenário da pesquisa (ALI e AWAD, 2014). As fontes de conhecimento são as cinco que foram identificadas por (REYNOLDS e SALEEM, 2005), estas são úteis na tomada de decisões (REYNOLDS e SALEEM, 2005; REYNOLDS e ALI, 2008):

- **Conhecimento situacional:** exemplos de soluções com sucesso e sem sucesso, etc.;
- **Conhecimento normativo:** intervalos de comportamentos aceitáveis;
- **Conhecimento de domínio:** conhecimento de objetos do domínio, suas relações e interações;
- **Conhecimento topográfico:** padrões espaciais de comportamento;
- **Conhecimento histórico:** padrões temporais de comportamento.

#### 2.3.6.1 *Conhecimento Situacional*

Esta fonte de conhecimento contém agentes exemplares de todos os tipos de soluções (bem e malsucedidas). A representação é uma estrutura com um conjunto de exemplares a partir da evolução da população. Nesta estrutura, cada exemplar tem um valor para cada parâmetro e o valor correspondente de fitness (ALI e AWAD, 2014).

#### 2.3.6.2 *Conhecimento Normativo*

Esta fonte de conhecimento pode ser entendida como limites aceitáveis para o comportamento do agente. A representação básica do conhecimento normativo é vista como um conjunto de intervalos, onde cada intervalo é uma faixa de promissores bons comportamentos (ALI e AWAD, 2014).

#### 2.3.6.3 *Conhecimento de Domínio*

Conhecimento de domínio orienta a busca através do estudo do domínio do problema e gerencia a informações incluídas a partir dele. Ele ajuda os indivíduos na exploração da sua busca e prevê padrões ambientais (ALI e AWAD, 2014).

#### 2.3.6.4 *Conhecimento de Topográfico*

Este conhecimento fornece comportamentos geograficamente notáveis dos agentes (padrões espaciais). É considerada um esquema regional que é representado como uma grade multidimensional que é composta por células (ALI e AWAD, 2014).

#### 2.3.6.5 *Conhecimento de Histórico*

Guarda o histórico de comportamentos dos agentes (padrões temporais). Nesta estrutura, o conjunto de todos os comportamentos de agente é coletado de parte da estrutura de conhecimento. Ele calcula a variação média de valores de parâmetros em uma determinada região e o período de tempo, e, em seguida, prevê a direção da mudança da posição anterior.

Este conjunto de categorias é visto como completo para um dado domínio, no sentido de que todo conhecimento disponível pode ser expresso em termos de uma dessas classificações (SILVA, 2012b).

### 2.3.7 *Protocolos de Comunicação*

São os protocolos de comunicação que organizam o conhecimento aceito e influência as futuras gerações de solucionadores de problemas (ALI *et al.*, 2014). O espaço populacional e o espaço de crenças são interligados através de duas principais funções:

- **A função Aceitação:** que dita às regras de seleção no espaço populacional;
- **A função Influência:** através da qual as culturas no espaço populacional são influenciadas por seus vizinhos aceitos no espaço de crença.

### 2.3.8 *Aplicação do Algoritmo Cultural*

O AC tem sido aplicado em diversas áreas desde sua apresentação ao mundo científico. O seu uso pode ser isolado ou em conjunto com outros algoritmos, onde estes trabalham de forma colaborativa na busca do melhor resultado para solução do problema abordado.

Por possuir em sua estrutura um espaço de crenças que agrupa cinco conhecimentos que são atualizados no decorrer de sua utilização, o AC tem sido utilizado nas atividades que necessitem de aprendizagem conforme forem executados. No trabalho de OSTROWSKI e REYNOLDS (1999), o AC foi utilizado para detecção de falhas em teste de software, onde inicialmente, este utilizou o teste da caixa preta aprendendo classes de equivalência de pares de entrada/saída defeituosas. Sendo em seguida, passado para outro AC as classes aprendidas, com o objetivo de identificar falhas específicas dentro do projeto do programa. O uso do AC na extração do conhecimento aplicado aos serviços e buscas na Web, é apresentado no trabalho de



REYNOLDS e STEFAN (2003), estes se utilizam da existência de muitos dados e da pobreza do conhecimento destes dados para utilizar a estrutura do AC como um mecanismo para evoluir e refinar as consultas de pesquisas significativas com o objetivo de extrair informações primárias e periféricas significativas e úteis.

No atual momento de avanço tecnológico, tem-se cada vez mais utilizado técnicas que necessitam da avaliação de grandes quantidades de dados e muitas vezes existe a necessidade de interpretação de dados coletados durante a realização de certas atividades, tais como, na automação e na robótica. Dentro deste contexto, WARIS e REYNOLDS (2018), apresenta em seu trabalho o uso do AC na distribuição de conhecimento de teoria dos jogos, suportando tanto a cooperação quanto a competição entre os jogadores.

A solução de problemas de otimização do sistema de energia, tem se mostrado uma área que cada vez mais temos técnicas evolutivas sendo proposta para encontrar melhores resultados para este tipo de problema. O AC, foi usado por diversos autores nesta vasta área (GOUDARZI *et al.*, 2017b; HEMMATI e AZIZI, 2017; VEERAMANI *et al.*, 2018).

De acordo com o exposto anteriormente, pode-se notar que o AC está sendo utilizado em diversas áreas aplicado em uma grande variedade de problemas.

## 2.4 Estratégias de Busca local

### 2.4.1 Heurísticas e Meta-heurísticas

A heurística é um método de busca ou otimização empírica de que muitas vezes se trabalha buscando-se apenas resolver o problema, mas não tem uma relação com qualquer prova matemática ou física rigorosa do que geralmente se espera. Ninguém sabe se a heurística vai sempre dar a melhor resposta para o problema. Não há mudanças de parâmetros e sim mudança de comportamento do código. A heurística é simplesmente usada como um atalho para resolver problemas difíceis. Segundo BECCENERI (2008), o termo heurístico é derivado da palavra grega *heuriskein*, que significa descobrir e que podemos considerar esse termo como associado a um conhecimento circunstancial, não verificável, nem matematicamente verificável.

A meta-heurística é uma estratégia de busca mais geral criada para um grupo de problemas que apresentam as mesmas condições e que você configura apenas os parâmetros para melhorar os resultados. Uma meta-heurística é uma estratégia de busca, não específica para um determinado problema, que tenta explorar eficientemente o espaço de soluções viáveis desse problema (BECCENERI, 2008).

A busca local é uma heurística de busca que usa uma operação definida **movimento** para definir uma vizinhança (ARMENTANO e BRANCHINI, 2013). Os algoritmos de busca local operam usando um único estado corrente (em vez de vários caminhos) e em geral se movem apenas para os vizinhos desse estado (KU e MAK, 1998). Sendo assim, se uma população já possui uma descendência melhorada onde estão as melhores possíveis soluções para o problema de otimização, com a busca local pode-se chegar ao melhor resultado. Os algoritmos de busca local são úteis para resolver problemas de otimização puros, nos quais o objetivo é encontrar o melhor estado de acordo com uma função objetivo (KU e MAK, 1998).

As estratégias mais utilizadas como busca local são: *Hill-Climbing*, *Simulated Annealing* e *Tabu Search* (SILVA, 2012b). Temos também o *Beam Search* que é um método do tipo Enumeração Implícita para resolver problemas de Otimização Combinatória (KRASNOGOR e SMITH, 2005).

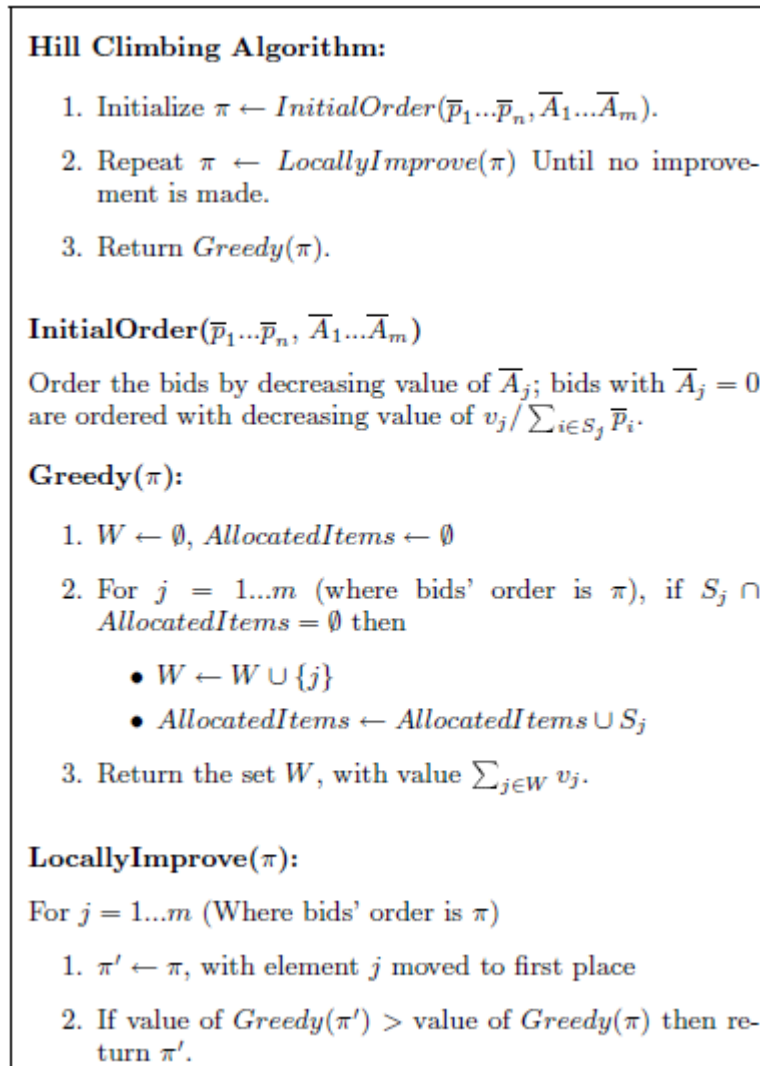
#### 2.4.2 Hill Climbing

Ele é simplesmente um laço repetitivo que se move de forma contínua no sentido do valor crescente – isto é, encosta acima (KU e MAK, 1998). O algoritmo básico (Ver Figura 2.4) desta estratégia pode ficar preso a máximos locais, mínimos locais ou a platôs, dependendo do espaço que está sendo explorado. A solução para este tipo de problema é utilizar os reinícios aleatórios (*random restart*), que consiste em realizar diversas buscas a partir de estados iniciais gerados aleatoriamente. Sendo que, cada busca será executada até que:

- Um número máximo de iterações definidas, seja realizado, ou;
- Não se encontrem melhoras significativas nos resultados obtidos.

É selecionado o melhor resultado encontrado com as diferentes buscas em diferentes reinícios.

**Figura 2.4:** Algoritmo básico *Hill Climbing*.



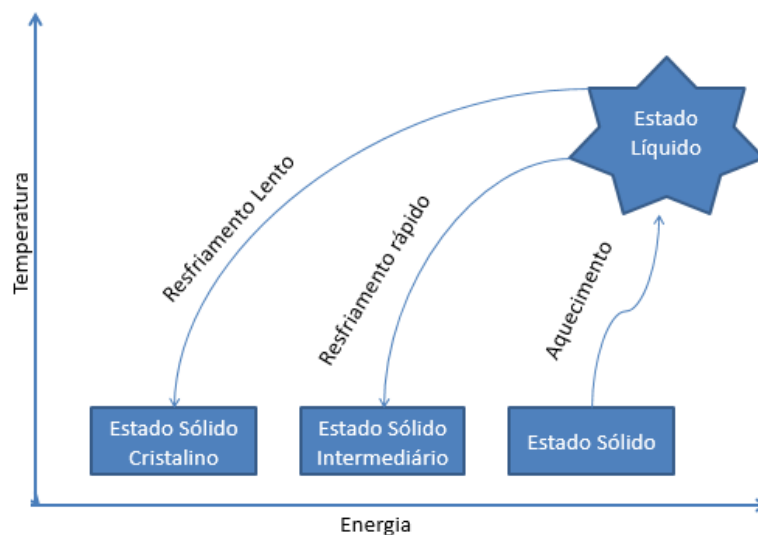
Fonte: (ZUREL e NISAN, 2001).

### 2.4.3 Simulated Annealing (SA)

Em metalurgia, a têmpera é o processo usado para temperar ou endurecer metais e vidro aquecendo-os a alta temperatura e depois esfriando-os gradualmente, permitindo assim que o material seja misturado em um estado cristalino de baixa energia (KU e MAK, 1998). O Recozimento Simulado é uma meta-heurística inspirada no processo físico de recozimento de um sólido para obtenção de estados de baixa energia na área da física da matéria condensada (GASPAR-CUNHA *et al.*, 2012). No Brasil tem-se usado a tradução de Recozimento simulado para a expressão *Simulated Annealing* (GASPAR-CUNHA *et al.*, 2012). O SA estabelece uma ligação entre esse tipo de comportamento termodinâmico e a busca de mínimos globais para um problema de otimização discreta. Dessa forma, o SA tornou-se uma busca local probabilística, com base na termodinâmica (HART, 1994).

A figura 2.5 exemplifica o processo de recozimento. O algoritmo de recozimento simulado é análogo ao processo físico de recozimento. Na tabela 2.1 é mostrada uma analogia entre o processo de recozimento e o algoritmo de recozimento simulado.

**Figura 2.5:** Processo de recozimento.



**Fonte:** Adaptado de (GASPAR-CUNHA *et al.*, 2012).

**Tabela 2.1:** Processo físico de recozimento versus o algoritmo de recozimento simulado.

Processo de Recozimento	Recozimento Simulado
Aplicado na busca pelo Equilíbrio Térmico da Matéria	Visa a Otimização de uma função
Variação energética	Variação da função objetivo
Procura determinar o estado de mínima energia	Procura determinar o valor mínimo da função objetivo
Baseado na diminuição da temperatura (T)	Baseado na diminuição do parâmetro de controle (c)
Pode parar em estados intermediários de energia da matéria, não atingindo o estado de mínima energia.	Pode parar em mínimos locais da função objetivo, não atingindo o mínimo global.

**Fonte:** Adaptado de (GASPAR-CUNHA *et al.*, 2012).

Da mesma forma que o sólido é resfriado lentamente para garantir uma estrutura cristalina, o algoritmo resfria a solução lentamente para garantir que ela tenha a melhor função objetivo ao mesmo tempo em que permite configurações que vão de encontro ao melhor valor da função objetivo encontrado (situação correspondente a pequenos aquecimentos) (SILVA, 2012b).

Segundo BECCENERI (2008), a ideia fundamental é permitir movimentos que resultem em soluções de pior qualidade do que a solução corrente, a fim de escapar de mínimos locais. A probabilidade de fazer um movimento desse tipo decresce durante a busca. A Figura 2.6, mostra o algoritmo do SA.

**Figura 2.6:** Algoritmo do SA.

<b>Algoritmo SimulatedAnnealing( )</b>	
01:	<b>INÍCIO</b>
02:	Pegar a solução S, a temperature inicial T e o fator de temperatura k
03:	melhor ← s;
04:	<b>REPITA ATÉ</b> (que melhor solução ideal, ou o tempo tenha esgotado, ou $T < 0$ )
05:	$S' \leftarrow$ GerarVizinho(S) /*aqui deve-se escolher um método de vizinhança*/
06:	$\Delta\text{custo} \leftarrow$ custo(S') - custo(S);
07:	<b>SE</b> ( $\Delta\text{custo} < = 0$ ) <b>ENTÃO</b>
08:	$S \leftarrow S'$ ;
09:	<b>SENÃO</b>
10:	$S \leftarrow S'$ com probabilidade $e^{-\Delta\text{custo}/T}$ ;
11:	$T \leftarrow K * T$ ; //redução de T com uma nova temperatura
12:	<b>SE</b> (S < melhor) <b>ENTÃO</b>
13:	melhor ← S;
14:	<b>FIMREPITA</b>
15:	<b>RETORNE</b> S (melhor solução)
16:	<b>FIM</b>

**Fonte:** Adaptado de (SILVA, 2012b).

#### 2.4.4 Tabu Search (TS)

Este método guia um procedimento heurístico de busca local pela utilização de características da solução corrente e da história da busca para explorar o espaço de soluções além da otimalidade local (ARMENTANO e BRANCHINI, 2013). Sendo utilizada como uma técnica de busca local, a TS parte de uma solução inicial e se move no espaço de soluções de uma solução para outra que esteja em sua vizinhança (GASPAR-CUNHA *et al.*, 2012). A TS é um procedimento de otimização poderoso que tem sido aplicado com sucesso a uma série de problemas combinatórios (KATSIGIANNIS *et al.*, 2012). A TS possui diversos parâmetros que compõe sua estrutura e segundo KATSIGIANNIS *et al.* (2012), para projetar um algoritmo TS é necessário, também, especificar os seguintes componentes básicos:

- (1) Critério de escolha da próxima solução vizinha;
- (2) Seleção dos atributos de movimento;
- (3) Memória de curto prazo para armazenar as regras de proibição (Lista Tabu);

(4) Número de iterações que o atributo selecionado permanecerá proibido (Tamanho da Lista Tabu);

(5) Critério de aspiração.

O uso sistemático de memória adaptativa constitui a propriedade que distingue busca tabu de outras meta-heurísticas. A palavra “adaptativa” significa que a memória atualiza o armazenamento de elementos de soluções ou soluções completas encontradas durante a exploração dos espaços de soluções (ARMENTANO e BRANCHINI, 2013).

Segundo ALI e AWAD (2014), a ideia básica do TS é baseada em um processo iterativo de busca na vizinhança em que um conjunto de movimentos começa a partir da solução opcional  $s$  com o objetivo de encontrar uma melhor solução  $s_0$  das regiões vizinhas  $N(s)$  até que um critério de parada seja cumprido.

O processo de intensificação é melhorado pelo uso das estruturas de memória, chamadas de listas tabu. Estas listas armazenam as soluções visitadas anteriormente ou um conjunto de regras fornecidas pelo usuário. A cada iteração é verificado se a solução corrente já foi visitada anteriormente ou se violou alguma regra, se sim, esta solução é armazenada na lista tabu e marcada como “tabu”. Este procedimento evita a chamada ciclagem, ou seja, que uma solução seja visitada novamente. Com esta estratégia de memória, o algoritmo TS pode ir além do ótimo local e acessar outras regiões do espaço de soluções (GASPAR-CUNHA *et al.*, 2012). Uma estrutura de fila tipo FIFO (*First IN First Out*) é usada para implementar a lista tabu, ou seja, quando o número de elementos é igual ao tamanho da lista e chega um novo elemento, então o primeiro que entrou sai da lista. Essa estratégia está fundamentada no fato de que na exploração do espaço de soluções, as soluções geradas a mais tempo, possivelmente estão “distantes” da região do espaço sob análise e, como tal, não tem influência na escolha da próxima solução vizinha naquela região (GASPAR-CUNHA *et al.*, 2012). O tamanho da lista tabu é considerado um parâmetro crítico. Pois segundo GASPAR-CUNHA *et al.* (2012), a dimensão da lista não pode ser tão pequena, sob pena de haver ciclagem; nem tão grande, para armazenar desnecessariamente soluções que não estejam ligadas à história recente da busca.

Outro parâmetro importante é o critério de aspiração. Segundo ARMENTANO e BRANCHINI (2013), os critérios de aspiração são usados em busca tabu para determinar quando regras de ativação tabu podem ser desconsideradas, o que permite que um movimento classificado como tabu possa ser executado. Alguns critérios de aspiração mais comuns, são apresentados por (GASPAR-CUNHA *et al.*, 2012):

**Aspiração por objetivo global:** Consiste em retirar o *status* tabu de um movimento se for produzida uma solução com a melhor avaliação global.

**Aspiração por objetivo regional:** Um movimento tabu perde seu *status* quando for gerada uma solução melhor que a melhor encontrada na região atual de busca.

**Aspiração Default:** Se todos os movimentos possíveis são tabu e não é possível aplicar outro critério de aspiração, então o movimento mais antigo perde sua condição de tabu.

Um algoritmo básico da técnica tabu é mostrado na Figura 2.7.

**Figura 2.7:** Algoritmo básico Tabu.

```

Begin
  initialize solution S = s0
  Set s* = s ;
  Set T=  $\emptyset$  ; // tabu list
  Initialize the aspiration function;
  iteration=0;
  best_iteration=0;
  While(iteration<iteration_max)
    iteration=iteration + 1;
    Create a set V* of solutions as:  $S_i = S + \Delta$  ,  $S_i$  does not belong to T;
    Find the best s' in V*;
    update T, f()aspiration;
    If((s') > f(s*)) Then
      Set s* = s' ;
      best_iteration=iteration;
    End
    Set s=s';
  End
End

```

**Fonte:** (ALI e AWAD, 2014).

#### 2.4.5 Beam Search (BS)

O algoritmo do *Beam Search* é um método do tipo Enumeração Implícita para resolver problemas de Otimização Combinatória (DE AZEVEDO *et al.*, 2013). Outra definição é apresentada por OHATA (2012), *Beam search* é um método heurístico para resolver problemas de otimização através de uma árvore de decisão, onde somente os nós mais promissores são mantidos a cada nível enquanto os demais são descartados permanentemente.

Este algoritmo inicia com k estados gerados aleatoriamente. Em cada passo, são gerados todos os sucessores de todos os k estados (árvore parcial chamada de *beam* que significa feixe), e se um destes for o objetivo, o algoritmo para. Caso contrário, são escolhidos a partir da lista completa os k melhores sucessores repetindo a ação. Para cada um desses feixes somente um nó descendente de cada nó pai é selecionado e os demais são descartados. Dessa forma,  $\beta$  (largura da busca) não são obtidos a cada nível da árvore de busca até que chegue ao último nível (OHATA, 2012).

Os nós que serão explorados no *beam search* são definidos por uma função avaliação ou estimativa. É informado por OHATA (2012), que dois tipos de função de avaliação têm sido utilizados: função de avaliação de prioridade e função de avaliação de custo total. Sendo que a primeira é de baixo custo computacional, mas não possuem precisão, pois podem descartar nós importantes. A segunda possui mais precisão, porém requerem maior esforço computacional.

## 2.5 Abordagem Memética

### 2.5.1 Considerações Iniciais

O desenvolvimento de algoritmos que não tem como base uma única metaheurística tradicional vem aumentando nos últimos anos. Porém nota-se que no início desta tendência à interação entre as técnicas utilizadas e entre as comunidades de pesquisa quase não existia. Este desenvolvimento combina diversos componentes provenientes de diversas áreas de pesquisa sobre otimização. Estas abordagens são comumente referidas como metaheurísticas híbridas (SILVA, 2012b). O conhecimento de diferentes áreas de otimização conduz na maioria das vezes ao desenvolvimento de uma abordagem híbrida eficaz. Porém mesmo com um conhecimento generalizado não é fácil o desenvolvimento de um algoritmo híbrido que consiga resolver todo tipo de problemas. Geralmente, alguns conseguem resolver certas classes de problemas. No entanto, existem vários tipos de hibridização que tem demonstrado ser bem-sucedidos para muitas aplicações e podem servir como orientação para novos desenvolvimentos (BLUM *et al.*, 2011).

Na busca de algoritmos mais eficientes na resolução de problemas de otimização, principalmente aqueles que não possuem informações específicas sobre eles, tem-se usado a combinação de metaheurísticas e heurísticas. Técnicas que empregam metaheurísticas como busca global e heurísticas específicas do problema como a busca local, são comumente referidos como Algoritmos Meméticos (AMs) (BLUM *et al.*, 2011). A denominação de “Algoritmos Meméticos” foi usada pela primeira vez em (MOSCATO, 1989). Segundo MOSCATO (1989), Algoritmo Memético é um casamento entre uma pesquisa global de base populacional e a busca local heurística feita por cada um dos indivíduos.

### 2.5.2 Implementação de Algoritmos Meméticos

Na implementação de um Algoritmo Memético (AM) não é exigido apenas um mecanismo de busca global juntamente com os operadores de pesquisa locais, mas também se



deve estabelecer uma coordenação sutil para expor o ponto de vista de ambos (SILVA, 2012b). Os algoritmos evolutivos híbridos são semelhantes aos algoritmos evolutivos tradicionais, a diferença está no fato que quando é incluído um procedimento de busca local que explore de forma extensiva a vizinhança em torno de um indivíduo, desta forma, o tradicional se torna híbrido.

Os algoritmos meméticos podem ser considerados Lamarckianos ou Baldwinianos, dependendo da abordagem de substituição da geração atual pela geração melhorada. Segundo SILVA (2012b)

- **AMs Baldwinianos:** quando o genótipo original do indivíduo selecionado for mantido. Neste caso, o cromossomo selecionado será associado à aptidão da busca local. Essa abordagem incrementa a diversidade, levando-se em consideração que no processo de seleção o indivíduo que tiver aptidão inferior terá chances de ser selecionado, desde que sua aptidão proveniente da busca local seja melhor (KU e MAK, 1998).

- **AMs Lamarckianos:** quando o resultado produzido pela busca local substitui o indivíduo selecionado na população (genótipo e aptidão). Essa abordagem pressupõe que os indivíduos podem passar geneticamente características de aprendizagem.

Os algoritmos meméticos levantam uma série de questões importantes que devem ser abordadas pelo desenvolvedor (KRASNOGOR e SMITH, 2005). Para SILVA (2012b), talvez a mais importante destas questões possa ser enunciada como: “Qual é o melhor equilíbrio entre a busca local e busca global no processo evolutivo?”. Em seu trabalho sobre problemas de otimização contínua (HART, 1994), fez alguns questionamentos sobre a concepção de algoritmos meméticos eficientes:

- Qual a frequência da aplicação da busca local?
- Quais indivíduos devem ser utilizados na busca local?
- Em quanto tempo deve ser executada a busca local?
- Qual a quantidade de eficiência que a busca local precisa ter?

A resposta a cada uma das questões acima conduz ao desenvolvimento de um AM mais eficiente e robusto.

#### **Frequência da aplicação da busca local**

Um estudo realizado com algoritmos genéticos e busca local realizado por HART (1994), concluiu que AGs com grandes populações são mais eficazes quando a busca local é utilizada com pouca frequência. Também afirma que quando o algoritmo não é capaz de identificar as regiões que contêm ótimos globais é necessária uma grande frequência de busca local.

### **Seleção dos indivíduos para aplicação da busca local**

A busca local pode ser aplicada a cada indivíduo de uma população, quando isso não é realizado, é necessário definir uma forma de selecionar os indivíduos para aplicar a busca local. Apesar da possibilidade de utilizar métodos mais sofisticados que usam informações da população para selecionar o melhor indivíduo, HART (1994) expõe que o método mais simples é o aleatório uniforme.

### **Tempo de execução da busca local**

Segundo SILVA (2012b), um AG híbrido com uma busca local de longa duração irá executar menos gerações do AG do que um AG híbrido com uma busca local de menor duração, se ambos terminarem com o mesmo número de avaliações de função. O uso de curta ou longa duração para busca local, depende do problema a ser resolvido. Isso é mostrado em HART (1994), que concluiu que o uso de um curto período de busca local produziu melhores resultados para as funções de Griewank, enquanto uma longa duração produziu melhores resultados para funções de Rastrigin. Muitas vezes o tempo de execução de uma busca local é limitado a um valor chamado profundidade de busca local (SILVA, 2012b).

### **Quantidade de eficiência que a busca local precisa ter**

Quando se seleciona um método de busca local para um modelo híbrido, o custo da busca local e sua eficiência são os dois fatores que podem afetar a eficiência híbrida global.

### **2.5.3 Aplicação do Algoritmo Memético**

Com o intuito de resolver problemas complexos os AMs utilizam a metaheurística associada com uma heurística. Desta forma, a metaheurística explora o domínio do problema realizando a busca global e a heurística específica do problema (busca local), intensifica a busca na vizinhança dos valores encontrados na pesquisa global.

No trabalho de ROSSI-DORIA e PAECHTER (2004), é apresentado um AM na solução do problema de agendamento de curso universitário. Este foi testado em instâncias de benchmark definidos para esta categoria de problemas, este obteve bons resultados e mostrou que a computação evolutiva com a ajuda da pesquisa local pode competir com algoritmos bem-sucedidos neste tipo de problema.

Um problema desafiador de otimização com múltiplos objetivos com decisões conflitantes e variáveis de decisão interdependentes, é a operação de controle de inundação de reservatórios (*Reservoir flood control operation - RFCO*). Na solução deste problema, QI *et al.* (2016), usou o AM inspirado no sistema imunológico com dois operadores de busca local, o primeiro sendo baseado na dominância de Pareto e o segundo inspirado por evolução

diferencial. Para medir o desempenho do RFCO, foram realizados estudos experimentais sobre problemas de benchmark. Para este tipo de problema, LUO *et al.* (2015), já tinha proposto a solução pelo uso de AM obtido pela hibridização da otimização do enxame de partículas com estimativa do algoritmo de distribuição.

O problema de roteamento de veículos com janelas de tempo (*vehicle routing problem with time windows* - VRPTW), é um problema de otimização discreta NP-difícil com dois objetivos: minimizar o número de veículos que atendem a um conjunto de clientes geograficamente dispersos e minimizar a distância total percorrida no plano de roteamento. Para solução deste tipo de problema, NALEPA e BLOCHO (2016), introduziram um algoritmo memético adaptativo denominado de AMA-VRPTW. Durante a busca os parâmetros do algoritmo vão se ajustando dinamicamente. Neste trabalho é aplicado o teste de *Wilcoxon* bicaudal para verificar a significância estatística dos resultados.

Os problemas de otimização de engenharia, geralmente são acompanhados de restrições que impossibilitam o uso de técnicas clássicas para solucioná-los. O artigo de PECHAC *et al.* (2016), apresentou uma implementação do algoritmo memético para otimização estrutural discreta. Este algoritmo era a combinação de algoritmo genético e cinco métodos de busca local: pesquisa de padrões, método simplex de *Nelder-Mead*, método de gradiente conjugado não-linear de Dai-Yuan (NCG), método de otimização de enxame de partículas (PSO) e método de projeto de estresse total. O objetivo era minimizar o preço do material usado, enquanto satisfaz as restrições de estresse e deslocamento. O algoritmo memético proposto era capaz de escolher o método correto de busca local baseado no desempenho que é avaliado em tempo real durante a otimização. O algoritmo foi testado em uma estrutura de treliça submetida a restrições de tensão e deslocamento. As combinações de melhor desempenho foram GA + PSO e GA + NCG, de modo que podem ser consideradas as melhores escolhas para solução de problemas semelhantes. A escolha de automatizar do método de pesquisa local não melhorou o desempenho neste problema de teste específico.

O problema de despacho econômico de energia, é considerado um problema de otimização multiobjetivo com restrições. Este faz parte dos problemas de engenharia do mundo real. Vários autores tem apresentados soluções pelo uso de AMs (BHUVANA e ARAVINDAN, 2016; NARANG *et al.*, 2017).

Pelo exposto acima pode-se observar que os algoritmos meméticos são utilizados com diversas configurações e em várias áreas. Estes são avaliados geralmente através da comparação do seu resultado, com os de outros trabalhos correlatos, pelo seu desempenho no uso de funções de benchmark e outras vezes por teste estatísticos baseados na média e no desvio padrão.

## 2.6 Técnicas para medidas de desempenho de algoritmos

O crescente interesse pela análise experimental no campo dos algoritmos evolutivos tem aumentado nos últimos anos. Diversos trabalhos abordam a análise e propostas de diferentes tipos de problemas que são utilizados como base para comparações experimentais de algoritmos. Além de proporem o uso de diferentes técnicas estatísticas e diferentes metodologias na comparação de algoritmos. Um perigo óbvio com a avaliação empírica de algoritmos de busca é que as conclusões resultantes dependem tanto de quais problemas são usados para teste quanto dos algoritmos que estão sendo comparados (WHITLEY *et al.*, 1996).

Nos cenários da computação evolucionária, geralmente tem-se que usar múltiplos algoritmos aplicados a múltiplos benchmarks, isso aumenta a dificuldade em classificar os algoritmos. É uma prática comum na computação evolutiva executar os algoritmos várias vezes e, em seguida, o valor médio e o desvio padrão são calculados. Para comparar o desempenho dos algoritmos, é muito comum o uso de testes estatísticos de hipóteses (KROHLING *et al.*, 2015).

Na avaliação de desempenho de algoritmos evolutivos, tem-se usado testes paramétricos e não-paramétricos. Os termos paramétrico e não-paramétrico referem-se à média e ao desvio-padrão, que são os parâmetros que definem as populações que apresentam distribuição normal (CAMPOS, 2001). A distinção entre estes dois tipos de testes para alguns autores, está na suposição específica com relação a um ou mais parâmetros populacionais em relação as distribuições subjacentes que as caracterizam em função do emprego dos testes a estes aplicados. A categorização de um procedimento como um teste paramétrico ou um não-paramétrico, pode se basear no nível de medição representado pelos dados a serem analisados. Para SHESKIN (2003), como regra geral, os testes estatísticos inferenciais que avaliam dados categóricos / nominais e dados ordinais de ordem de lote são categorizados como testes não paramétricos, enquanto os testes que avaliam dados de intervalo ou dados de razão são categorizados como testes paramétricos.

Uma dúvida comum é a de qual tipo de método utilizar, paramétrico ou não-paramétrico? CAMPOS (2001), criou um diagrama que esquematiza as subdivisões dos testes estatísticos mais utilizados na prática (Tabela 2.2).

**Tabela 2.2:** Testes estatísticos paramétricos e não-paramétricos.

Testes Estatísticos			
Paramétricos		Não-Paramétricos	
Independentes	Vinculados	Independentes	Vinculados
2 amostras	2 amostras	2 amostras	2 amostras
Teste <i>t</i> (Student)	Teste <i>t</i> (Student)	Mann-Whitney	Wilcoxon
		T. da Mediana	T. dos sinais
Mais de duas	Mais de duas	$\chi^2$ (2 x 2)	Mac Nemar
		Proporções Exato (Fisher)	Binomial
Análise de variância	Análise de variância	Mais de duas	Mais de duas
		Kruskal-Wallis	
		Mediana (m x n)	Cochran
		$\chi^2$ (m x n)	Friedman
		Nemenyi	

Fonte: (CAMPOS, 2001).

Pela estrutura da tabela 4.1, pode-se identificar o teste a ser utilizado de acordo com as características das variáveis de cada abordagem.

### 2.6.1 Métodos paramétricos

Segundo BARNDORFF-NIELSEN (2012), o conceito central de inferência paramétrica é o de probabilidade.

A verificação de algumas condições auxilia na definição do uso dos testes paramétricos. Estas verificações são: a realização de testes que julgam a normalidade da distribuição dos erros amostrais, homogeneidade das variâncias, aditividade dos efeitos provocados pelos fatores de variação sobre a variável e a independência dos erros. Para CAMPOS (2001), se a distribuição não for normal, se não houver homogeneidade das variâncias ou se os efeitos não forem aditivos, existem duas alternativas: 1) ou tentar uma transformação dos dados originais; ou então 2) utilizar testes que não levam em conta os parâmetros amostrais (média e desvio-padrão), ou seja, usar a estatística por isso mesmo chamada não-paramétrica. A primeira alternativa é a opção que se deve tentar utilizar antes de

partir para a segunda alternativa, porque a estatística paramétrica é mais poderosa do que a não paramétrica (CAMPOS, 2001; SHESKIN, 2003). Existem diversas transformações diretas dos dados que podem ser utilizadas, entre estas podemos destacar: a logarítmica, a logarítmica dos (dados+1), a transformação dos valores de z, a transformação angular, a raiz quadrada dos dados, a raiz quadrada dos (dados + 1, ou mais  $\frac{1}{2}$ ), a raiz cúbica dos dados, a transformação hiperbólica de primeiro grau (ou o inverso dos dados) ou hiperbólica de segundo grau e a transformação percentual.

Segundo SHESKIN (2003), há um consenso geral entre a maioria dos pesquisadores de que, desde que não haja razão para acreditar que uma ou mais suposições de um teste paramétrico tenham sido violadas, quando o nível de medição para um conjunto de dados for intervalo ou razão, os dados devem ser avaliados com o teste paramétrico apropriado. Este comenta ainda que, embora os testes paramétricos geralmente forneçam um teste mais poderoso de uma hipótese alternativa do que seus análogos não-paramétricos, a vantagem do poder de um teste paramétrico pode ser negada se uma ou mais suposições forem violadas.

### 2.6.2 Métodos não paramétricos

Uma análise estatística paramétrica pode não ser apropriada, especialmente quando lidamos com resultados de múltiplos problemas. Na análise de múltiplos problemas, propomos o uso de testes estatísticos não paramétricos, uma vez que são menos restritivos que os paramétricos e podem ser usados em amostras de tamanho pequeno de resultados. Eles podem ser usados para analisar dados nominais e reais, por meio do uso de medidas baseadas em classificação. São ferramentas poderosas para a análise de resultados em Inteligência Computacional (GIBBONS e CHAKRABORTI, 2011). A necessidade em definir o comportamento de algoritmos quando submetidos a problemas de diferentes naturezas, tem aberto um campo de pesquisa em procedimentos de testes (GARCÍA *et al.*, 2009; DERRAC *et al.*, 2014). A seguir são apresentados alguns testes não-paramétricos aplicados na avaliação do comportamento de algoritmos, em diversos cenários.

### 2.6.3 Teste de Friedman

O teste de Friedman é um teste de comparações múltiplas que visa detectar diferenças significativas entre o comportamento de dois ou mais algoritmos (DERRAC *et al.*, 2011). O processo para realização do teste de Friedman segue os passos a seguir, segundo (DERRAC *et al.*, 2011):

- 1- Reunir todos os resultados de cada par de algoritmo/problema;

- 2- Classificar os valores de cada problema  $i$  de 1 (melhor resultado) até  $k$  (pior resultado). Note esta classificação como  $r_j^i$  ( $1 \leq j \leq k$ );
- 3- Para cada algoritmo  $j$ , calcular a média das classificações obtidas em todos os problemas para obter a classificação final  $R_j = \frac{1}{n} \sum_i r_i^j$ .

Desta forma os algoritmos são classificados para cada problema separadamente. Como indicado no item 2, o algoritmo com melhor desempenho é classificado com 1, o segundo melhor com 2, etc.

A estatística de Friedman ( $F_f$ ) é calculada de acordo com a equação 2.1.

$$F_f = \frac{12n}{k(k+1)} \left[ \sum_j R_j^2 - \frac{k(k+1)^2}{4} \right], \quad (2.1)$$

Na expressão (4.1),  $n$  é o número de problemas, sendo  $i$ , o índice associado ao número de problemas.  $k$  é o número de algoritmos e o  $j$ , é o índice associado a este.

#### 2.6.4 Teste de Friedman Aligned

Para o teste de Friedman alinhado ( $F_{AR}$ ), é calculado um valor de localização como o desempenho médio alcançado por todos os algoritmos em cada problema. O passo de obter a diferença entre o desempenho de um algoritmo e o valor da localização é repetida para cada combinação de algoritmos e problemas. A equação 2.2, mostra a definição para o cálculo estatístico da classificação alinhada de Friedman ( $F_{AR}$ ).

$$F_{AR} = \frac{(k-1) \left[ \sum_{j=1}^k \hat{R}_j^2 - (kn^2/4)(kn+1)^2 \right]}{\{[kn(kn+1)(2kn+1)]/6\} - (1/k) \sum_{i=1}^n \hat{R}_i^2}, \quad (2.2)$$

Na expressão (2.2),  $n$  é o número de problemas, sendo  $i$ , o índice associado ao número de problemas.  $k$  é o número de algoritmos e o  $j$ , é o índice associado a este.

#### 2.6.5 Teste de Quade

O teste de Quade é o terceiro teste utilizado neste trabalho (o primeiro é o teste de Friedman e o segundo é o teste de Friedman alinhado). Este teste difere do de Friedman (que considera todos os problemas iguais em termos de importância), porque leva em conta o fato que alguns problemas são mais difíceis do que outros ou que as diferenças registradas na

execução de vários algoritmos são maiores. Portanto, os rankings calculados em cada problema podem ser dimensionados dependendo das diferenças observadas nos desempenhos dos algoritmos, obtendo, como resultado, uma análise de classificação ponderada da amostra (DERRAC *et al.*, 2011).

O teste de Quade ( $F_Q$ ) pode ser calculado pela equação 2.3, levando em consideração algumas definições apresentadas em (DERRAC *et al.*, 2011). Considerando também os termos A e B, dados pelas equações 2.4 e 2.5, respectivamente.

$$F_Q = \frac{(n-1)B}{A-B}, \quad (2.3)$$

Os valores de A e B, foram assumido para simplificar a função original apresentada em (DERRAC *et al.*, 2011). Seus valores são apresentados abaixo:

$$A = n(n+1)(2n+1)k(k+1)(k-1)/72, \quad (2.4)$$

$$B = \frac{1}{n} \sum_{j=1}^k k S_j^2, \quad (2.5)$$

Nas expressões (2.4) e (2.5),  $n$  é o número de problemas, sendo  $i$ , o índice associado ao número de problemas.  $k$  é o número de algoritmos e o  $j$ , é o índice associado a este. E  $S_j$  é a soma da média de classificação de cada algoritmo.

Os testes discutidos nesta seção, foram e são utilizados em diversas áreas e situações. Na área de computação, THEODORSSON-NORHEIM (1987), aplicou estes testes para avaliar um programa em BASIC para analisar dados biomédicos de resposta de sujeitos experimentais a um estímulo, monitorado em intervalos de tempo. Estes testes continuam sendo usados em conjunto ou separadamente para avaliar algoritmos aplicados a diversos problemas (ODA *et al.*, 2014; BENA VOLI *et al.*, 2016).

### 2.6.6 Hellinger-TOPSIS

TOPSIS (Technique for Order Preference by Similarity to Ideal Solution – Técnica para ordenar preferências pela similaridade para a solução ideal) é amplamente usado para tratar problemas de tomada de decisões no mundo real e tem sido generalizado para lidar com uma variedade de tipos de informação (BEHZADIAN *et al.*, 2012). Este é o quarto teste que será utilizado neste trabalho. O uso desta técnica para medição de desempenho de algoritmos



evolucionários exige uma adaptação assumindo certas premissas, visto que é comum a utilização de testes de hipóteses estatísticas para comparar esses tipos de algoritmos (DERRAC *et al.*, 2011). A adaptação do TOPSIS com a distância de Hellinger utiliza o conceito do Teorema do Limite Central, para reduzir o número de hipóteses. Em geral tem-se uma matriz de decisão  $D$  com alternativas e critérios (KROHLING *et al.*, 2015), de acordo com a expressão (2.6).

$$D = \begin{matrix} & C_1 & \cdots & C_n \\ \begin{matrix} A_1 \\ \cdots \\ A_m \end{matrix} & \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \end{matrix} \quad (2.6)$$

Onde,

$A_1, A_2, \dots, A_m \rightarrow$  São Alternativas

$C_1, C_2, \dots, C_n \rightarrow$  São Critérios

$x_{mn} \rightarrow$  Indica a classificação da alternativa  $A_m$  de acordo com os critérios  $C_n$

A matriz decisão pode ser representada conforme apresenta a Figura 2.8.

**Figura 2.8:** Exemplo de tabela para a matriz de decisão.

	Critério 1	Critério 2	Critério 3	Critério $n$
Alternativa 1	Distribuição de Probabilidade			
Alternativa 2				
Alternativa 3				
Alternativa $n$				

Fonte: Autores, (2017).

Onde as alternativas são os algoritmos a serem ranqueados e os critérios são as funções de benchmarks.

Os passos a seguir são de acordo com (KROHLING *et al.*, 2015). O primeiro passo é identificar o PIS (Positive Ideal Solutions – Soluções ideais positivas), ou seja, os benefícios e o NIS (Negative Ideal Solutions – Soluções ideais negativas), ou seja, os custos (KROHLING *et al.*, 2015). Isto é feito para cada critério (Funções de Benchmarks), para um problema de minimização utiliza-se para PIS o menor valor da média e para o NIS o maior valor da média correspondente. Ou seja:

$$PIS - f_j^+ \leftarrow f_{ij} \text{ onde } i \text{ é o índice correspondente ao } \min_i(\mu_{ij}) \forall j=1, \dots, n \text{ e } \forall_i = 1, \dots, m.$$

$$NIS - f_j^- \leftarrow f_{ij} \text{ onde } i \text{ é o índice correspondente ao } \max_i(\mu_{ij}) \forall j=1, \dots, n \text{ e } \forall_i = 1, \dots, m.$$

Com  $f_{ij} \sim N(\mu_{ij}, \sigma_{ij})$  onde  $\mu_{ij}, \sigma_{ij}$  são a média e o desvio padrão, respectivamente.

O segundo passo é calcular a medida de separação  $d_i^+$  do PIS ( $f^+$ ) e  $d_i^-$  do NIS ( $f^-$ ) para cada alternativa. De acordo com (KROHLING *et al.*, 2015), como os benchmarks possuem o mesmo peso, os cálculos se tornam mais simples e são realizadas através das equações (2.7) e (2.8) respectivamente.

$$d_i^+ = \sum_{j=1}^n D_H(f_j^+, f_{ij}) \text{ com } i = 1, \dots, m. \quad (2.7)$$

$$d_i^- = \sum_{j=1}^n D_H(f_j^-, f_{ij}) \text{ com } i = 1, \dots, m. \quad (2.8)$$

O terceiro passo é o cálculo do coeficiente de proximidade relativa  $\xi_i$  para cada alternativa (Algoritmo) com relação ao PIS correspondente, como visto na equação (2.9).

$$\xi_i = \frac{d_i^-}{d_i^+ + d_i^-} \quad (2.9)$$

O quarto passo é realizar a ordenação das alternativas de acordo com o valor encontrado no coeficiente de proximidade. Sendo que as melhores alternativas são as que possuem maiores valores de coeficiente.

A técnica de medição de múltiplos algoritmos apresentada nesta seção foi utilizada por alguns autores (BEHZADIAN *et al.*, 2012; OUENNICHE *et al.*, 2018).

### 3 METODOLOGIA APLICADA

Nesta seção serão apresentados os procedimentos metodológicos aplicados nesta tese, que se baseia em quatro fases:

1. Pesquisa bibliográfica sobre algoritmos culturais, genéticos, meméticos, heurísticas de busca local, meta-heurísticas, otimização de funções multimodais, solução de problemas de otimização em engenharia e desenvolvimento de software para as análises realizadas;
2. Desenvolvimento do algoritmo memético com a hibridização do algoritmo cultural e quatro heurísticas utilizadas na busca local: Tabu Search, Beam Search, Hill Climbing e Simulated Annealing;
3. Avaliação do algoritmo desenvolvido na utilização de funções de benchmark pelo uso de testes não paramétricos;
4. Avaliação do algoritmo memético na aplicação em problemas de engenharia.

#### *3.1 Pesquisa bibliográfica*

Nesta fase, foi realizada uma revisão bibliográfica sobre as pesquisas relacionadas com o estudo de algoritmos meméticos desenvolvidos, seus componentes e aplicações, o uso de algoritmos em problemas de otimização de problemas de variáveis reais e o estado da arte neste assunto. Os dados de entrada para esta pesquisa serão obtidos através de publicações de artigos e pesquisas científicas divulgadas dentro do Brasil e pelo mundo. Revistas indexadas, periódicos científicos e sites tais como: IEEE Explorer, Web of Knowledge, Engineering Village, Scopus SciVerse, Springer Link, entre outros serão pesquisados.

### 3.2 *Desenvolvimento do algoritmo memético*

Nesta fase foram desenvolvidas as classes de busca local previstas nos objetivos específicos desta tese. Além de ter melhorado o algoritmo cultural que faz parte do *framework* (Anexo) utilizado por SILVA (2012b) que foi baseado no JCLEC (<http://jclec.sourceforge.net/>), com a inclusão do conhecimento topográfico. Nesta fase, a contribuição desta tese ficou caracterizada pela forma de alimenta as informações para o conhecimento topográfico.

### 3.3 *Avaliação do algoritmo desenvolvido na aplicação de funções de benchmark*

O algoritmo desenvolvido e suas variações foram aplicados na solução de funções de *benchmark* e depois avaliados pelo uso de testes não paramétricos.

Nesse trabalho, foram criados alguns cenários de simulação com o objetivo de medir o desempenho dos ACs e suas adaptações com busca local. Os parâmetros modificados para cada simulação foram: Tamanho da população (Tam Pop), Número de gerações (Num Ger) e Número de repetições (Num Rep). A quantidade máxima de avaliações que é o produto do Tam Pop por Num Ger, foi mantido no valor de 10.000 e 30.000 avaliações por cada repetição para funções dependendo do cenário a ser utilizado.

#### **3.3.1 Funções de benchmark**

As funções de benchmarks são bastante utilizadas quando se deseja medir o desempenho de algoritmos. Essas funções são utilizadas para comparar o Algoritmo Cultural (AC) e quatro propostas de hibridização, AC com Simulated Annealing (SA), AC com Busca Tabu (BT), AC com *Hill Climbing* (HC) e AC com *Beam Search* (BS).

Os cinco algoritmos foram testados em oito funções de benchmarks do CEC2017 (AWAD *et al.*, 2016). As definições de espaço de busca  $[-100.0, 100.0]^D$  e das dimensões  $D=10$  e  $D=30$ . Sendo que a primeira avaliação foi realizada pela técnica Hellinger-TOPSIS em quatro funções básicas (tabela 3.1). A segunda avaliação, foi aplicada em quatro funções básicas conforme mostra a tabela 3.1 e em quatro funções híbridas conforme mostra a tabela 3.2, pelo uso dos testes de Friedman, Friedman Aligned e Quade.

**Tabela 3.1:** Funções Básicas de benchmark.

Função	Descrição e Expressão	
F1	Função Bent Cigar	$f_1(x) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2$ ,
F3	Função Zakharov	$f_3 = \sum_{i=1}^D x_i^2 + (\sum_{i=1}^D 0.5x_i)^2 + (\sum_{i=1}^D 0.5x_i)^4$ ,
F5	Função Rastrigin's	$f_5(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$
F15	Função Griewank's	$f_{15}(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ ,

Fonte: Adaptada de (AWAD *et al.*, 2016).

**Tabela 3.2:** Funções Híbridas de benchmark.

Função	Dom. Funções Básicas	Descrição	Funções Básicas
FH1	P = [0.2, 0.4, 0.4]	Função híbrida 1	g1: Função Zakharov g2: Função Rosenbrock g3: Função Rastrigin's g1: Função Elíptica condicionada Alta
FH4	P = [0.2, 0.2, 0.2, 0.4]	Função híbrida 4	g2: Função Ackley's g3: Função Schaffer's g4: Função Rastrigin's g1: Função Bent Cigar
FH5	P = [0.2, 0.2, 0.3, 0.3]	Função híbrida 5	g2: Função HGBat g3: Função Rastrigin's g4: Função Rosenbrock g1: Função Elíptica condicionada Alta g2: Função Ackley's
FH8	P = [0.2, 0.2, 0.2, 0.2, 0.2]	Função híbrida 8	g3: Função Rastrigin's g4: Função HGBat g3: Função Discus

Fonte: Adaptada de (AWAD *et al.*, 2016).

### 3.3.2 Avaliação pela técnica Hellinger-TOPSIS

Esta avaliação foi realizada pelo uso das quatro funções básicas de benchmark (tabela 3.1), com as etapas:

#### A. Cenário 1 e 2

Para estes dois cenários foi atribuído o valor do Número de Repetições igual 50. As tabelas 3.3 e 3.4, apresentam os parâmetros dos cenários 1 e 2, respectivamente. Sendo que a diferença entre estes dois cenários está em relação ao número de variáveis utilizadas. Para o cenário 1, foram utilizadas 10 variáveis, enquanto que para o cenário 2 foram utilizadas 30 variáveis.

**Tabela 3.3:** Cenário 1.

ID	Tam Pop	Num Ger	Max Avaliação	Num Rep
F1-10	50	200	10.000	50
F3-10	50	200	10.000	50
F5-10	50	200	10.000	50
F15-10	50	200	10.000	50

**Tabela 3.4:** Cenário 2.

<b>ID</b>	<b>Tam Pop</b>	<b>Num Ger</b>	<b>Max Avaliação</b>	<b>Num Rep</b>
<b>F1-30</b>	50	600	30.000	50
<b>F3-30</b>	50	600	30.000	50
<b>F5-30</b>	50	600	30.000	50
<b>F15-30</b>	50	600	30.000	50

### B. Variações no Algoritmos

Com a combinação das variações dos parâmetros foi possível obter 7 (sete) alternativas (algoritmos). A tabela 3.5, apresenta cada uma das alternativas. Sendo que, o AC foi utilizado na sua estrutura básica, enquanto na variação do AC+SA foram mantidos os parâmetros do AC e a variável de Energia de Busca Local sofreu alterações de 5, 10 e 15. Criando assim 3 novas condições de testes para este algoritmo híbrido. Na variação do AC+BT, também foram mantidos os parâmetros do AC, porém houve variação no tamanho da lista tabu em 2, 4 e 6. Novamente, criando mais 3 condições de testes. Para o AC+HC, foram mantidos os parâmetros do AC e mantido o valor constante igual a 10, a variável que define o tamanho da planície. Na hibridização do AC+BS, os parâmetros do AC continuaram constantes, porém, a variável que define a quantidade de feixes, sofreu variações 4, 8 e 12. Desta forma, foram criadas um total de 11 condições de testes.

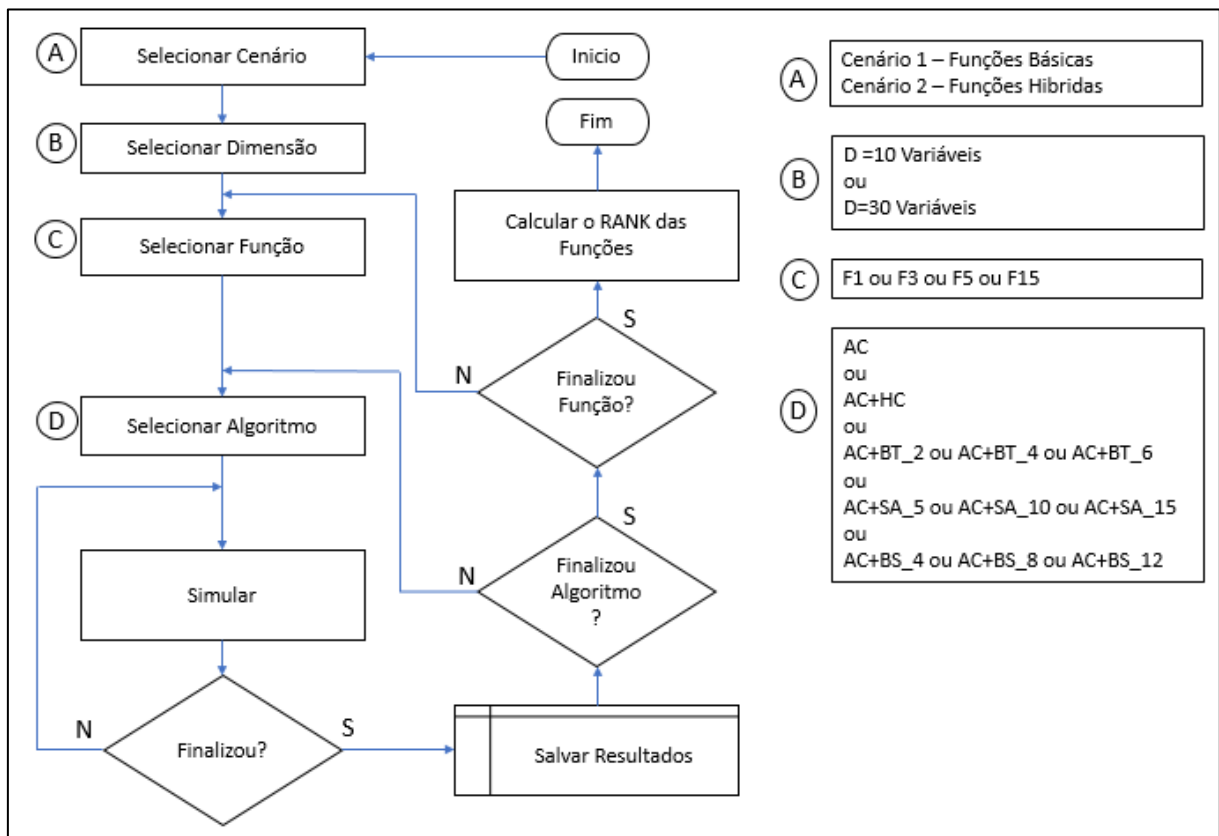
**Tabela 3.5:** Algoritmos para teste.

<b>Algoritmo</b>	<b>Descrição</b>
<b>AC</b>	Algoritmo Cultural Clássico
<b>AC+BS_4</b>	Algoritmo Cultural + <i>Beam Search</i> (Quantidade de Feixes = 4)
<b>AC+BS_8</b>	Algoritmo Cultural + <i>Beam Search</i> (Quantidade de Feixes = 8)
<b>AC+BS_12</b>	Algoritmo Cultural + <i>Beam Search</i> (Quantidade de Feixes = 12)
<b>AC+HC</b>	Algoritmo Cultural + Hill Climbing
<b>AC+SA_5</b>	Algoritmo Cultural + <i>Simulated Annealing</i> (Energia de Busca Local = 5)
<b>AC+SA_10</b>	Algoritmo Cultural + <i>Simulated Annealing</i> (Energia de Busca Local = 10)
<b>AC+SA_15</b>	Algoritmo Cultural + <i>Simulated Annealing</i> (Energia de Busca Local = 15)
<b>AC+BT_2</b>	Algoritmo Cultural + Busca Tabu (Tamanho da Lista Tabu = 2)
<b>AC+BT_4</b>	Algoritmo Cultural + Busca Tabu (Tamanho da Lista Tabu = 4)
<b>AC+BT_6</b>	Algoritmo Cultural + Busca Tabu (Tamanho da Lista Tabu = 6)

### C. Fluxograma para o teste

A figura 3.1, mostra através de um fluxograma as etapas utilizadas para execução dos testes para este artigo.

**Figura 3.1:** Organograma para o teste.



### 3.3.3 Avaliação pelas técnicas de Friedman, Friedman Aligned e Quade

Esta avaliação foi realizada pelo uso das quatro funções básicas (tabela 3.1) e das quatro funções híbridas de benchmark (tabela 3.2), com as etapas:

#### A. Cenário 1 e 2

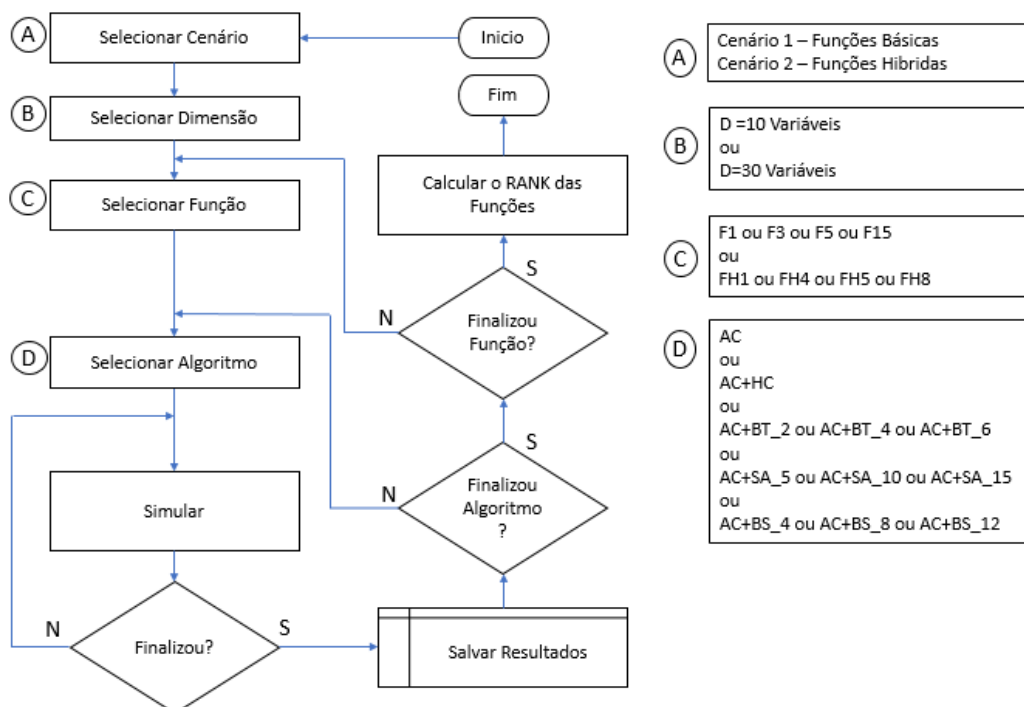
Os dois cenários possuem Número de Repetições igual a 50. A cada nova repetição é gerada uma nova população aleatória com função densidade de probabilidade uniforme definida dentro dos limites de cada variável. A tabela 3.6 apresenta os parâmetros dos cenários 1 e 2, relacionados com funções básica e híbridas respectivamente.

**Tabela 3.6:** Cenário 1 e 2 – Funções Básicas e Híbridas.

D	Cenário1	Nome Função	Cenário2	Nome Função	Tam Pop.	Num. Ger	Max. Avaliação	D*Max_FES	Num Rep
10	F1	Bent Cigar	FH1	Híbrida 1	50	200	10.000	100.000	50
10	F3	Zakharov	FH4	Híbrida 4	50	200	10.000	100.000	50
10	F5	Rastrigin's	FH5	Híbrida 5	50	200	10.000	100.000	50
10	F15	Griewank's	FH8	Híbrida 8	50	200	10.000	100.000	50
30	F1	Bent Cigar	FH1	Híbrida 1	50	600	30.000	300.000	50
30	F3	Zakharov	FH4	Híbrida 4	50	600	30.000	300.000	50
30	F5	Rastrigin's	FH5	Híbrida 5	50	600	30.000	300.000	50
30	F15	Griewank's	FH8	Híbrida 8	50	600	30.000	300.000	50

### B. Fluxograma para o teste

A figura 3.2, mostra através de um fluxograma as etapas utilizadas para execução dos testes para este artigo.

**Figura 3.2:** Organograma para o teste.

Para todos os ACs utilizados, manteve-se o mesmo padrão de seus parâmetros e foram realizados testes preliminares com a intenção de definir o tamanho da variável de energia do (AC+SA), o tamanho da lista tabu do (AC+BT) e a quantidade de feixes do (AC+BS), e após estes testes ficou definido que três variáveis de cada técnica que se mostrassem com resultados



mais diferenciados, seriam utilizadas para incrementar novos cenários de testes. Na variação do AC com o SA, a variável de Energia (SA) de Busca Local sofreu alterações de 5, 10 e 15. Na variação do AC com o BT, ocorre a variação do tamanho da lista tabu em 2, 4 e 6. Na variação do AC com o BS, a variação do feixe fincou para 4, 8 e 12. Dessa maneira, surgem dez novas condições de testes para este algoritmo híbrido, acrescentando o AC+HC. Ficando, então com onze variações de algoritmos (Tabela 3.5).

### *3.4 Avaliação do algoritmo memético na aplicação em problemas de engenharia*

Os problemas de otimização de engenharia têm diferentes variáveis de design, possuem restrições complexas e possuem características não lineares. As restrições podem ser vistas como duas classes: a) Restrições devido ao intervalo permitido das variáveis do projeto, e b) restrições devido às condições características do problema. As restrições são muito importantes nos problemas de projeto de engenharia, uma vez que são geralmente impostas na declaração de problemas e às vezes muito difíceis de serem satisfeitas, o que torna a pesquisa difícil e ineficiente (BRAJEVIC *et al.*, 2010). A não linearidade, na maioria dos casos, leva a uma abordagem multimodal e algoritmos metaheurísticos podem ser aplicados (BEHESHTI e SHAMSUDDIN, 2013; GOGNA e TAYAL, 2013; YANG *et al.*, 2014; NAJI, 2017).

#### **3.4.1 Engenharia Elétrica**

O despacho econômico de carga e emissões utiliza as variáveis de custo de combustível e emissão de gases de forma minimizada para obter uma operação ótima em unidades de geração em uma usina de energia, garantindo a oferta de demanda. A primeira variável é definitiva para garantir a continuidade dos negócios e a segunda para cumprir a legislação ambiental e não degradação do meio ambiente. O algoritmo proposto neste trabalho foi aplicado na solução de um problema de despacho econômico de carga, utilizando dados de uma usina real com 10 geradores e o sistema do IEEE com 13 unidades geradoras, na situação clássica, que opera com todos os geradores buscando minimizar o custo e a emissão atendendo a demanda especificada; na situação controlada, são desligados os geradores que têm o maior custo incremental de combustível, mas garantem a demanda e reduzem a emissão de gases. Os resultados obtidos foram comparados entre si e com os resultados de outras técnicas relatadas

na literatura. Além da planta real, foi modelado no algoritmo proposto o sistema de 13 geradores do IEEE.

#### A. DESPACHO ECONÔMICO DE CARGA E EMISSÃO COMBINADOS

Despacho econômico de carga (ELD) é uma das principais tarefas de otimização em sistemas de energia. O principal objetivo da ELD é determinar a distribuição ótima da demanda de energia entre as unidades geradoras comprometidas, com a minimização do custo operacional total, ao mesmo tempo em que satisfaz um conjunto de restrições de igualdade e desigualdade. Por causa da preocupação ambiental, uma das restrições que devem ser levadas em conta é a importância das restrições ambientais (ZHU, 2015).

A avaliação de variáveis relacionadas ao custo do combustível e a emissão de gases são consideradas importantes quando se trata do despacho de carga em usinas termelétricas. As funções que dependem dessas variáveis são minimizadas para garantir um ótimo CEELD (Combined Economic Emission Load Dispatch), as funções objetivo são: custo e ambiental, respectivamente.

##### - Objetivo de custo

A função objetivo que deve ser minimizada para manter o custo total de combustível do sistema é modelada pela função quadrática dada por (3.1) (SECUI, 2015). Onde,  $a_i$ ,  $b_i$  e  $c_i$  são os coeficientes de custo de combustível de cada unidade geradora,  $n$  é o número de geradores e  $P_i$  a potência ativa de cada gerador.

$$F_1(P_i) = \sum_{i=1}^n (a_i + b_i P_i + c_i P_i^2) \quad \$/h \quad (3.1)$$

##### - Objetivo ambiental

Para a emissão de gases, a função objetivo também é modelada por uma função quadrática dada por (3.2) (ZIANE *et al.*, 2014). Onde  $d_i$ ,  $e_i$  e  $f_i$  são os coeficientes de emissão.

$$F_2(P_i) = \sum_{i=1}^n (d_i + e_i P_i + f_i P_i^2) \quad \text{kg/h} \quad (3.2)$$

A minimização das funções  $F_1(P)$  e  $F_2(P)$  no conjunto de possíveis soluções de energia geradas por cada gerador, é utilizada na definição do problema de otimização multiobjetivo através de:

$$\text{Minimize } P [F_1(P), F_2(P)] \quad (3.3)$$

A definição dada pela expressão acima está diretamente relacionada às restrições que acompanham o problema de otimização multiobjetivo aplicado ao CEELD. Para este trabalho, são consideradas as restrições de igualdade do equilíbrio de poder e as restrições de desigualdade em termos de capacidade de geração. Uma restrição de equilíbrio de poder de igualdade é a soma algébrica dos poderes do sistema em estudo deve ser nula (ALTUN e YALCINOZ, 2008) (SECUI, 2015). Esta restrição é dada por (3.4),

$$\sum_{i=1}^n P_i - P_D - P_L = 0 \quad (3.4)$$

onde  $P_i$ , é a potência de saída de cada gerador,  $P_D$  é a demanda de potência e  $P_L$  são as perdas de potência associadas à transmissão.

As perdas da linha de transmissão  $P_L$ , dada por (3.5), em todo o sistema, são funções quadráticas em relação às variáveis  $P_j$  e são calculadas com a fórmula do coeficiente B (SECUI, 2015):

$$P_L = \sum_{i=1}^N \sum_{j=1}^N P_i B_{ij} P_j + \sum_{i=1}^M B_{0i} P_i + B_{00} \quad (3.5)$$

onde  $B_{ij}$  é um elemento da matriz de coeficientes de perda de tamanho  $n \times n$ ,  $B_{0i}$  é o elemento  $i$  do vetor de coeficientes de perda de tamanho  $n$  e  $B_{00}$  o coeficiente de perda constante (SECUI, 2015). A restrição de desigualdade em termos de capacidade de geração: para ter uma operação suave e estável do sistema, todos os geradores são fortemente limitados para operar com seus limites mínimo e máximo de geração (GOUDARZI *et al.*, 2017a), dado por (3.6):

$$P_{min.i} \leq P_i \leq P_{max.i} \quad (3.6)$$

Onde:

$P_i$  - Potência de saída do gerador  $i$

$P_{min.i}$  - Gerador de energia mínima  $i$

$P_{max.i}$  - Potência máxima do gerador  $i$

## B. PLANTA REAL DA GERAÇÃO DE ENERGIA

Os dados da usina e seu conjunto de geradores foram obtidos de (JÚNIOR *et al.*, 2017), que usaram geradores a gás 10 J620 (fabricante: GE Jenbacher, Jenbach - Áustria). Eles usaram um processo de pesquisa de campo para coletar os dados necessários para gerar as Tabelas 3.7

e 3.8, que mostra as características de cada gerador que opera nesta usina em relação aos coeficientes para funções de custo de combustível quadrático e funções de emissão.

**TABELA 3.7:** Dados característicos dos geradores do estudo de caso.

Gerador	$c_i$ (\$/Mw <sup>2</sup> )	$b_i$ (\$/Mw)	$a_i$ (\$)	$P_{min}$ (Mw)	$P_{max}$ (Mw)
PG1	0.007	7	240	0.66	3.35
PG2	0.0095	10	200	0.9	3.7
PG3	0.009	8.5	220	0.8	3.6
PG4	0.009	11	200	0.66	3.35
PG5	0.008	10.5	220	0.72	3.45
PG6	0.0075	12	120	0.66	2.97
PG7	0.0075	14	130	0.88	3.5
PG8	0.0075	14	130	0.754	3.33
PG9	0.0075	14	130	0.9	3.9
PG10	0.0075	14	130	0.56	2.35

Fonte: (NASCIMENTO *et al.*, 2016).

**TABELA 3.8:** Matriz de emissões dos geradores do estudo de caso.

Gerador	$f_i$ (mg/m <sup>3</sup> h)/MW <sup>3</sup>	$e_i$ (mg/m <sup>3</sup> h)/MW <sup>2</sup>	$d_i$ (mg/m <sup>3</sup> h)/
PG1	0.00419	1.32767	73.85932
PG2	0.00419	0.32767	13.85932
PG3	0.00683	-0.54551	40.2669
PG4	0.00683	-0.54551	40.2669
PG5	0.00461	-0.51116	42.89553
PG6	0.00461	-0.51116	42.8955
PG7	0.00461	-0.51116	42.8955
PG8	0.00461	-0.51116	42.8955
PG9	0.00061	-0.51116	10.8955
PG10	0.00461	-0.51116	42.8955

Fonte: (JÚNIOR *et al.*, 2017).

Os coeficientes de perda ( $B_m$ ) são dados por uma matriz quadrada de tamanho  $n \times n$ , onde  $n$  é o número de motores. A tabela 3.9, mostra a matriz de perdas.

**TABELA 3.9:** Matriz de perdas dos geradores da usina (Valores devem ser multiplicados por 1e-4).

n	1	2	3	4	5	6	7	8	9	10
1	0.14	0.17	0.15	0.19	0.26	0.22	0.34	0.38	0.43	0.45
2	0.17	0.60	0.13	0.16	0.15	0.20	0.23	0.56	0.23	0.51
3	0.15	0.13	0.65	0.17	0.24	0.19	0.25	0.38	0.43	0.45
4	0.19	0.16	0.17	0.71	0.30	0.25	0.43	0.56	0.23	0.51
5	0.26	0.15	0.24	0.30	0.69	0.32	0.18	0.37	0.42	0.48
6	0.22	0.20	0.19	0.25	0.32	0.85	0.97	0.55	0.27	0.58
7	0.34	0.23	0.25	0.43	0.18	0.97	0.67	0.38	0.43	0.45
8	0.38	0.56	0.38	0.56	0.37	0.55	0.38	0.56	0.23	0.51
9	0.43	0.23	0.43	0.23	0.42	0.27	0.43	0.23	0.42	0.48
10	0.45	0.51	0.45	0.51	0.48	0.58	0.45	0.51	0.48	0.45

O AC+SA foi selecionado para realizar a comparação do híbrido com o algoritmo SA clássico utilizado por (JÚNIOR *et al.*, 2017). O AC+BT foi utilizado por existir o uso da busca

tabu na solução de outros problemas ELD (LIN *et al.*, 2002; POTHYA *et al.*, 2008). O problema a ser resolvido pela AC Multi-Objetivo com a otimização da Pesquisa Local (SA ou BT) pode ser formulado da seguinte forma como a Equação (3.3), onde:

- O  $F_1(P_i)$  é a equação de custo do combustível do  $i_{th}$  motor, é o custo do combustível (\$) *versus* a energia gerada (MW). Os valores dos coeficientes  $a_i, b_i, c_i$ , e cada  $P_i$  são obtidos a partir da Tabela 3.7. Normalmente, é expressa por (3.1) a equação quadrática contínua;
- O  $F_2(P_i)$  é a equação de emissão do  $i_{th}$  motor, é a emissão (Kg / h) *versus* a potência gerada (MW). Os valores dos coeficientes  $d_i, e_i, f_i$ , são obtidos da Tabela 3.8. Normalmente, é expressa por (3.2) a equação quadrática contínua.

As perdas de energia são calculadas por (3.5) e as restrições usadas neste caso são (3.7) e (3.8).

$$P_{min.i} \leq P_i \leq P_{max.i}, i \in N_G \quad (3.7)$$

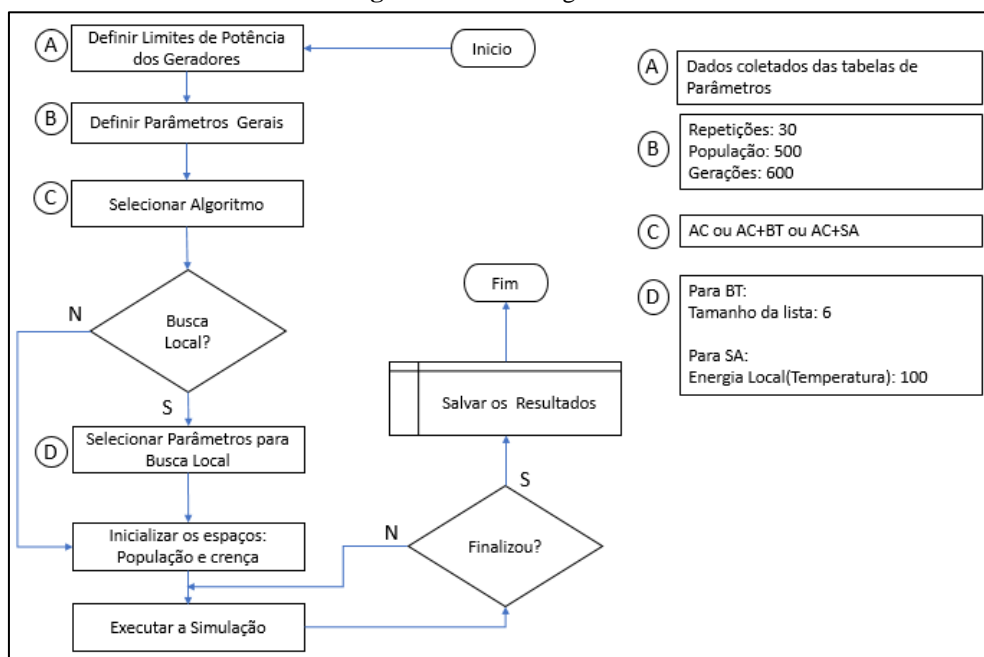
onde  $N_G$  refere-se a um número total de geradores.

$$\sum_{i=1}^n P_i - P^D - P^L = 0 \quad (3.8)$$

O algoritmo para determinar o CEELD de uma usina de energia usando AC multiobjetivo com o método de Busca Local (SA ou TS) é descrito no pseudocódigo da Figura 4.4. Os parâmetros de problemas são preenchidos no arquivo xml.

Os Parâmetros do problema são: Restrições do Gerador (Potência Máxima e Potência Mínima); Número de repetições; tamanho da população; Número de Gerações; Seleção do algoritmo a ser utilizado (AC, AC + SA ou AC + BT); Tabu List Size; E energia local de pesquisa para SA. A Figura 3.3, mostra o fluxograma do método.

Figura 3.3: O fluxograma do método.



### C. SISTEMA DE TESTE IEEE ELD COM 13 UNIDADES

Os dados nas Tabelas 3.10 e 3.11 são do sistema de teste IEEE de 13 unidades (RAJASOMASHEKAR e ARAVINDHABABU, 2012). Nelas está a informação das características de cada gerador em relação aos coeficientes para funções de custo de combustível quadrático e funções de emissão.

O sistema foi submetido aos mesmos testes utilizados no item B, com uma demanda programada para 2520 MW.

**Tabela 3.10:** Dados dos geradores do Sistema de Teste IEEE de 13 Unidades.

Geradores	$c_i$ (\$/Mw <sup>2</sup> )	$b_i$ (\$/Mw)	$a_i$ (\$)	$P_{min}$ (Mw)	$P_{max}$ (Mw)
PG1	0.00028	8.10	550	0.0	680.0
PG2	0.00056	8.10	309	0.0	360.0
PG3	0.00056	8.10	307	0.0	360.0
PG4	0.00324	7.74	240	60.0	180.0
PG5	0.00324	7.74	240	60.0	180.0
PG6	0.00324	7.74	240	60.0	180.0
PG7	0.00324	7.74	240	60.0	180.0
PG8	0.00324	7.74	240	60.0	180.0
PG9	0.00324	7.74	240	60.0	180.0
PG10	0.00284	8.60	126	40.0	120.0
PG11	0.00284	8.60	126	40.0	120.0
PG12	0.00284	8.60	126	55.0	120.0
PG13	0.00284	8.60	126	55.0	120.0

**Tabela 3.11:** Matriz de Emissões do Sistema de Teste IEEE de 13 Unidades.

Geradores	$f_i$ (mg/m <sup>3</sup> h)/MW <sup>3</sup>	$e_i$ (mg/m <sup>3</sup> h)/MW <sup>2</sup>	$d_i$ (mg/m <sup>3</sup> h)/
PG1	0.0632	-2.434	40
PG2	0.0348	-3.63	50
PG3	0.0348	-3.63	50
PG4	0.0438	-5.271	40
PG5	0.0438	-5.271	40
PG6	0.0438	-5.271	40
PG7	0.0438	-5.271	40
PG8	0.0438	-5.271	40
PG9	0.0438	-5.271	40
PG10	0.0571	-4.852	100
PG11	0.0571	-4.852	100
PG12	0.0571	-4.852	100
PG13	0.0571	-4.852	100

As tabelas 3.10 e 3.11 são usadas como entrada para novas simulações usando os mesmos parâmetros usados no sistema real de geração de energia com 10 geradores, aplicando os algoritmos AC, AC+SA e AC+BT na solução do problema CEELD.

### 3.4.2 Engenharia Mecânica

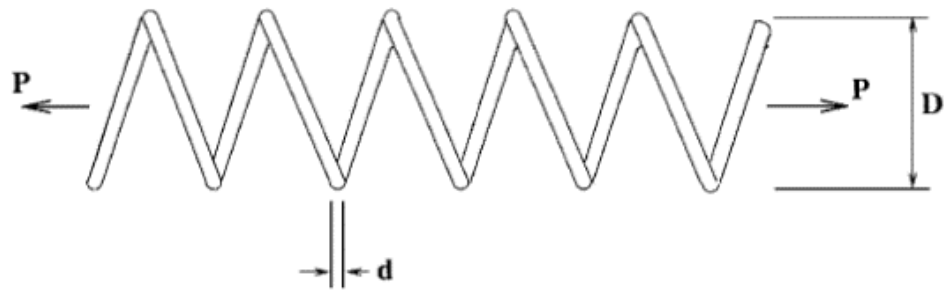
Dentre os problemas de otimização em engenharia mecânica foram selecionados três problemas de otimização com graus de dificuldades variados, o primeiro foi o projeto da mola helicoidal de compressão, que apresenta três variáveis e quatro restrições; o segundo foi o projeto do vaso sob pressão, que tem quatro variáveis e quatro restrições, e o quarto foi o projeto do redutor de velocidade, que apresenta sete variáveis e onze restrições. Os resultados obtidos em todos os problemas abordados foram comparados com o trabalho de outros autores e mostraram-se melhores.

Foram adotados para estes problemas: 10 iterações, população de 50 indivíduos, 300 gerações e tamanho da lista tabu de 6.

#### A. Projeto de Mola Helicoidal de Compressão

Este problema tem três variáveis e sete restrições (três restrições relacionadas ao intervalo permitido de suas variáveis e quatro restrições ligadas às características físicas). A figura 3.4 apresenta a mola de compressão helicoidal.

**Figura 3.4:** A mola helicoidal de compressão.



**Fonte:** (SILVA, 2012a).

O objetivo é minimizar a função de peso da mola de compressão helicoidal, dada por (3.9):

$$f(x) = (x_3 + 2)x_2x_1^2, \quad (3.9)$$

Sujeito a:

$$g_1 = 1 - (x_2^3x_3/71785x_1^4) \leq 0, \quad (3.10)$$

$$g_2 = \frac{4x_2^2 - x_1x_2}{12566x_2x_1^3 - x_1^4} + \frac{1}{5108x_1^2} - 1 \leq 0, \quad (3.11)$$

$$g_3 = 1 - (140.45x_1/x_2^2x_3) \leq 0, \quad (3.12)$$

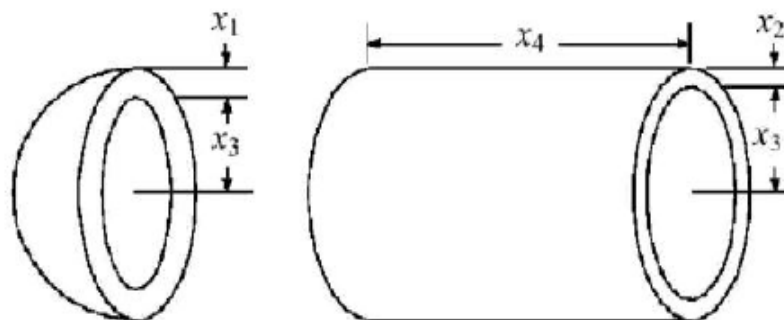
$$g_4 = ((x_1 + x_2)/1.5) - 1 \leq 0. \quad (3.13)$$

com  $0.05 \leq x_1 \leq 2.0$ ,  $0.25 \leq x_2 \leq 1.3$ , e  $2.0 \leq x_3 \leq 15.0$ .

## B. Projeto do vaso de pressão

problema tem quatro variáveis e oito restrições (quatro restrições relacionadas à faixa de variação permitida de suas variáveis e oito restrições ligadas a características físicas). A figura 3.5 apresenta o vaso de pressão.

**Figura 3.5:** Vaso de pressão.



**Fonte:** (CAGNINA *et al.*, 2008).



Este problema visa minimizar o custo total, incluindo custo do material, modelagem e soldagem, de um vaso cilíndrico, que é limitado em suas extremidades por cabeças hemisféricas (SILVA, 2012a). A função objetivo é dada por (3.14):

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (3.14)$$

Sujeito a:

$$g_1 = -x_1 + 0.0193x_3 \leq 0, \quad (3.15)$$

$$g_2 = -x_2 + 0.00954x_3 \leq 0, \quad (3.16)$$

$$g_3 = -\pi x_3^2 x_4^2 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0, \quad (3.17)$$

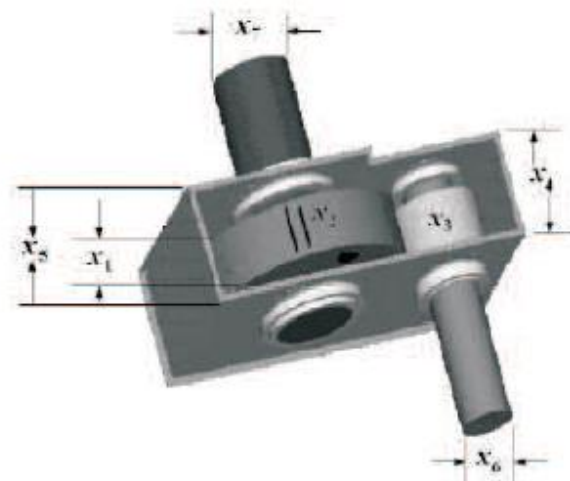
$$g_4 = x_4 - 240 \leq 0 \quad (3.18)$$

com  $1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625, 10.0 \leq x_3, x_4 \leq 200.0$ .

### C. Redutor de Velocidade

Esse problema tem sete variáveis e dezoito restrições (sete restrições relacionadas ao intervalo permitido de suas variáveis e onze restrições vinculadas a características físicas). A figura 3.6, apresenta o redutor de velocidade.

**Figura 3.6:** Redutor de velocidade.



Fonte: (BRAJEVIC *et al.*, 2010).

O peso do redutor de velocidade deve ser minimizado, sujeito a restrições na tensão de flexão dos dentes da engrenagem, tensão superficial, deflexões transversais dos eixos e

tensões no eixo (BRAJEVIC *et al.*, 2010). A função objetiva que modela o peso é dada por (3.19):

$$\begin{aligned}
 f(x) = & 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) \\
 & -1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) \\
 & +0.7854(x_4x_6^2 + x_5x_7^2)
 \end{aligned} \tag{3.19}$$

Sujeito a:

$$g_1 = (27/x_1x_2^2x_3) - 1 \leq 0, \tag{3.20}$$

$$g_2 = (397.5/x_1x_2^2x_3^2) - 1 \leq 0, \tag{3.21}$$

$$g_3 = (1.93x_4^3/x_2x_3x_6^4) - 1 \leq 0, \tag{3.22}$$

$$g_4 = (1.93x_5^3/x_2x_3x_7^4) - 1 \leq 0, \tag{3.23}$$

$$g_5 = \frac{1.0}{110x_6^3} \sqrt{\left(\frac{745.0x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0, \tag{3.24}$$

$$g_6 = \frac{1.0}{110x_7^3} \sqrt{\left(\frac{745.0x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6} - 1 \leq 0, \tag{3.25}$$

$$g_7 = (x_2x_3/40) - 1 \leq 0, \tag{3.26}$$

$$g_8 = (5x_2/x_1) - 1 \leq 0, \tag{3.27}$$

$$g_9 = (x_1/12x_2) - 1 \leq 0, \tag{3.28}$$

$$g_{10} = ((1.5x_6 + 1.9)/x_4) - 1 \leq 0, \tag{3.29}$$

$$g_{11} = ((1.1x_7 + 1.9)/x_5) - 1 \leq 0, \tag{3.30}$$

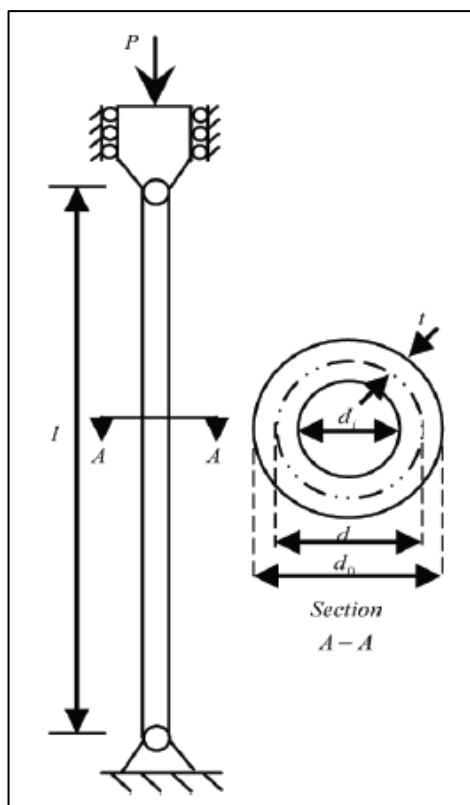
com  $2.6 \leq x_1 \leq 3.6$ ,  $0.7 \leq x_2 \leq 0.8$ ,  $17 \leq x_3 \leq 28$ ,  $7.3 \leq x_4 \leq 8.3$ ,  $7.8 \leq x_5 \leq 8.3$ ,  $2.9 \leq x_6 \leq 3.9$ , e  $5.0 \leq x_7 \leq 5.5$ .

### 3.4.3 Engenharia Civil

#### A. Projeto de coluna tubular

Esse problema tem duas variáveis e quatro restrições (duas restrições relacionadas ao intervalo permitido de suas variáveis e duas restrições ligadas às características físicas). A Figura 3.7 mostra a coluna tubular uniforme, Tabela 3.12, mostra os dados para o projeto.

**Figura 3.7:** Coluna Tubular uniforme.



Fonte: (GANDOMI e ALAVI, 2015).

Tabela 3.12: Dados para o projeto.

Descrição	Símbolo	Valor	Unidade
Carga	P	2500	kgf
Tensão de escoamento	$\sigma_y$	500	kgf/cm <sup>2</sup>
Módulo de Elasticidade	E	850000	kgf/cm <sup>2</sup>
Densidade	$\rho$	0.0025	kgf/cm <sup>3</sup>
Comprimento da Coluna	L	250	cm
PI	$\pi$	3.1415926	

O objetivo é minimizar o custo da coluna que é composta pelos custos de materiais e construção. A função objetivo para minimização é dada por (3.31).

$$f(d, t) = 9.8dt + 2d \quad (3.31)$$

Sujeito a:

$$g_1 = (P/\pi dt\sigma_y) - 1 \leq 0; \quad (3.32)$$

$$g_2 = (8PL^2/\pi^3 Edt(d^2 + t^2)) - 1 \leq 0. \quad (3.33)$$

onde:  $2 \leq d \leq 14$  e  $0.2 \leq t \leq 0.8$ .

## 4 ALGORITMO MEMÉTICO CULTURAL

### 4.1 *Considerações Iniciais*

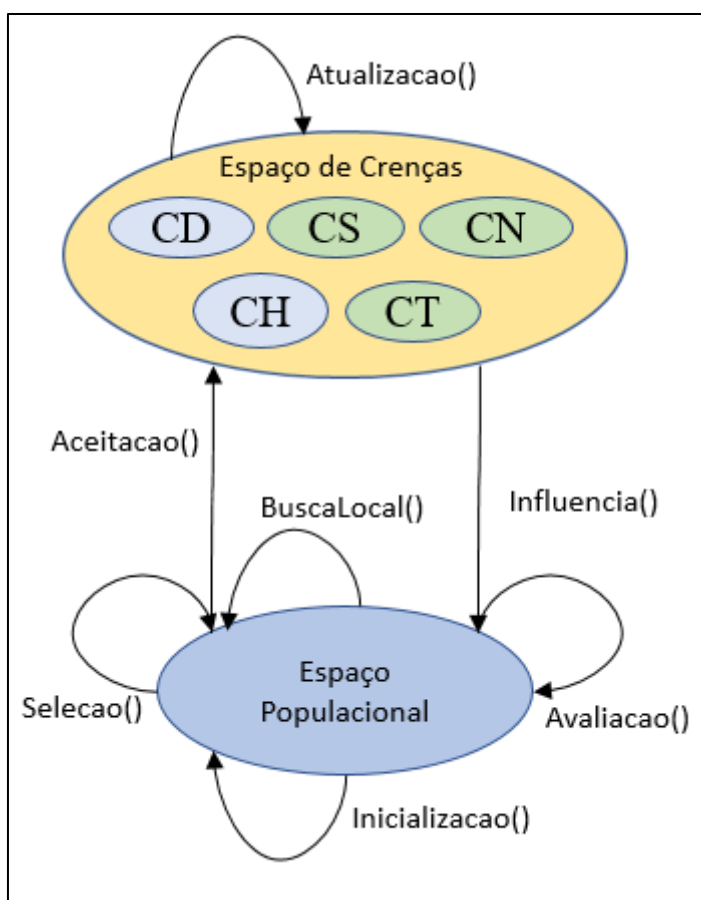
A hibridização de algoritmos evolucionários com heurísticas de busca local tem sido muito aplicada para solucionar diversos problemas (SHAHOOKAR *et al.*, 1994; MATSUMURA *et al.*, 2000; SUN *et al.*, 2009; NAITALI e GIRI, 2010; KATSIGIANNIS *et al.*, 2012; ALI e AWAD, 2014). O uso do algoritmo cultural também vem sendo divulgado e usado em vários trabalhos científicos (SUN *et al.*, 2009; ZHANG, 2011; ALI *et al.*, 2014; CHEN *et al.*, 2014). De forma a contribuir com este vasto campo da ciência o trabalho foi proposto e implementado em um ambiente criado com base no algoritmo cultural com população gerada pelo algoritmo genético, um Algoritmo Memético Cultural com Busca Local – AMCBL, onde foram desenvolvidas as classes utilizando as heurísticas de busca local com os algoritmos *Tabu Search*, *Beam Search*, *Simulated Annealing* e *Hill Climbing*. No decorrer deste capítulo serão descritas a interação destas classes com o algoritmo cultural.

### 4.2 *Melhorias no Algoritmo Cultural*

O algoritmo cultural, possui dois principais componentes, como descrito na seção 2.3.3 deste trabalho. Ou seja, o espaço populacional e o espaço de crenças.

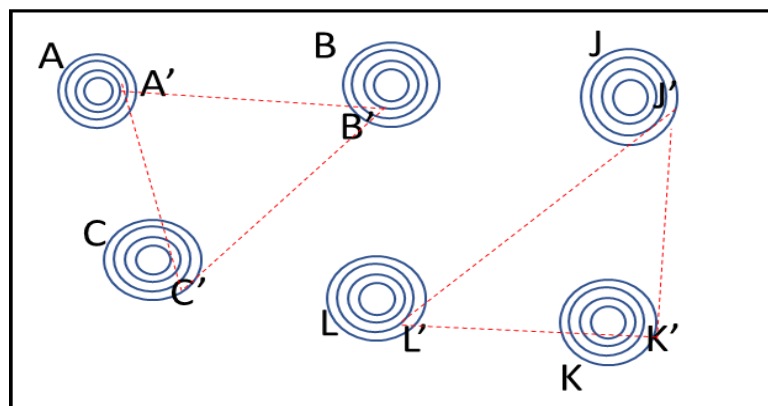
No espaço populacional temos uma população gerada pelo algoritmo genético que em algumas situações servirá para delimitar o domínio do problema a ser solucionado, e outras vezes poderá ser utilizado como entrada para satisfazer alguma restrição do problema que será resolvido. Já o espaço de crenças, é o local onde fica armazenado os cinco conhecimentos: situacional (CS), de domínio (CD), histórico (CH), normativo (CN) e topográfico. Porém, neste trabalho os conhecimentos atualizados foram o situacional, normativo e o topográfico, em verde na figura 4.1.

**Figura 4.1:** Algoritmo cultural com destaque nos conhecimentos deste trabalho.



O uso das informações dos três melhores indivíduos encontrados na busca local é utilizado para definir uma área de bom comportamento que alimenta o conhecimento topográfico. Essa abordagem até então não havia sido utilizada, de acordo com a pesquisa bibliográfica realizada. A figura 4.2, mostra um exemplo de áreas desmembradas demarcadas como melhores resultados na pesquisa local.

**Figura 4.2:** Áreas separadas de melhores resultados da busca local.

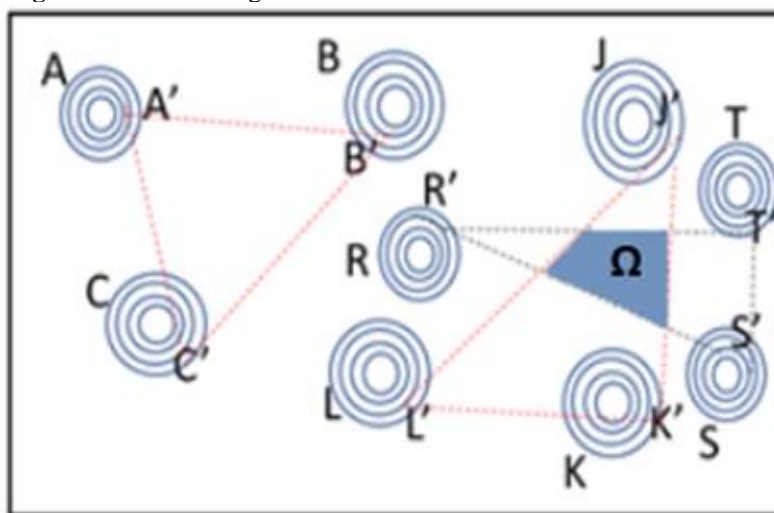


Os pontos A', B' e C' foram os melhores valores encontrados quando a busca local foi realizada na vizinhança de A, B e C, respectivamente. Assim, esses valores demarcam uma área que será aceita para conhecimento topográfico. O mesmo acontece com os pontos J', K' e L', que indicam os melhores valores nas vizinhanças de J, K e L. Antes de atualizar o conhecimento topográfico com essa nova informação, verifica-se se existe uma intersecção entre a área armazenada e esta nova área. Neste caso, a intersecção é o conjunto vazio, ou seja,  $(\Delta A'B'C') \cap (\Delta J'K'L') = \emptyset$ , portanto, as duas áreas são consideradas promissoras e são armazenadas no conhecimento topográfico para influenciar novos indivíduos dependendo de uma probabilidade definida como um parâmetro de entrada.

Após uma nova busca local, foram encontrados os pontos R', S' e T', que definem o melhor local de busca na vizinhança de R, S e T, gerando uma nova área marcada pelo triângulo  $\Delta R'S'T'$ . Neste caso, esta área tem uma região em comum com o triângulo  $\Delta J'K'L'$ , isto é  $(\Delta R'S'T') \cap (\Delta J'K'L') = \Omega$ . A nova área é armazenada no conhecimento topográfico juntamente com o resultado da intersecção obtida  $\Omega$ , que passa a ser uma área altamente promissora, que por influenciar os novos indivíduos têm um aumento na probabilidade definida no parâmetro de entrada.

A figura 4.3, mostra um exemplo com áreas congruentes demarcadas após uma pesquisa local.

**Figura 4.3:** Áreas congruentes de melhores resultados da busca local.



Os demais conhecimentos foram atualizados de forma clássica. As técnicas de busca local foram aplicadas dentro da função de mutação. A figura 4.4, mostra o pseudocódigo do algoritmo proposto.

**Figura 4.4:** Pseudocódigo do algoritmo proposto.

```

01: INÍCIO
02: variáveis:  $g=0$  (geração atual);  $g_{uv}=0$  (geração do último valor ótimo),
03:  $dm$  (diferença mínima para chamar a BuscaLocal)
04: inicializar população  $P(t)$  //população inicial aleatória
05: Inicializar Espaço de Crença  $EP(t)$ 
06: avaliar população  $P(t)$  // calcula  $f(i)$  para cada indivíduo
07: ENQUANTO (não condição_fim) FAÇA
08:   diferença= $g - g_{uv}$ ; // o valor de tu dependerá da evolução
09:   Comunicação ( $P(t)$ ,  $EP(t)$ ); // votação (Aceitação)
10:   Atualização  $EP(t)$ ; // uso de operadores culturais
11:   SE (diferença $\geq dm$ ) ENTÃO
12:     Selecionar Melhor_Individuo da população  $P(t)$ ;
13:     Melhor_Encontrado, Area_3_Melhores  $\leftarrow$  BuscaLocal(Melhor_Individuo);
14:     Comunicação (Melhor_Encontrado( $t$ ), Area_3_Melhores( $t$ ),  $EP(t)$ );
15:     Atualização  $EP(t)$ ; // uso de operadores culturais
16:   FIMSE
17:   Comunicação ( $EP(t)$ ,  $P(t)$ ); // promoção (função de influência)
18:    $t \leftarrow t+1$ ; // próxima geração
19:   selecionar  $P(t)$  de  $P(t-1)$ ;
20:   altera  $P(t)$ ; // crossover e mutação
21:   avaliar  $P(t)$ ; // calcula  $f(i)$  para cada indivíduo
22:   SE (diferença $\geq dm$ ) ENTÃO
23:     Selecionar Individuo_Aleatório da população  $P(t)$ ;
24:     Melhor_Encontrado, Area_3_Melhores  $\leftarrow$  BuscaLocal(Individuo_Aleatório);
25:     Comunicação (Melhor_Encontrado( $t$ ), Area_3_Melhores( $t$ ),  $EP(t)$ );
26:     Atualização  $EP(t)$ ; // uso de operadores culturais
27:   FIMSE
28: FIMENQUANTO
29: FIM

```

### 4.3 Hibridização do Algoritmo Cultural com a Busca Local

Neste trabalho foi escolhido o algoritmo cultural como metaheurística, devido a característica de rápida evolução de seus indivíduos no contexto cultural. Pois, a cada execução o espaço de crenças recebe novas informações que alimentam os conhecimentos utilizados no problema abordado. Para se aproximar cada vez mais da melhor solução, é realizada uma busca local que intensifica a pesquisa na vizinhança do possível melhor resultado.

Foram utilizados como busca local os algoritmos da busca tabu (*Tabu Search*), subindo a colina (*Hill Climbing*), recozimento simulado (*Simulated Annealing*) e a busca em feixes (*Beam Search*). Cada hibridização do algoritmo cultural com a busca local selecionada, possui parâmetros específicos.

### 4.3.1 Algoritmo Cultural com Busca Tabu (AC+BT)

A formulação da busca tabu, pode ser vista como:

$$\begin{aligned} \text{Min } f(x) \\ \text{Sujeito a: } x \in X \end{aligned} \quad (4.1)$$

Onde  $x$  é um conjunto de configurações ou as variáveis de decisão, e  $f(x)$  é a função objetivo, sendo  $X$  o espaço de busca (universo de configurações).

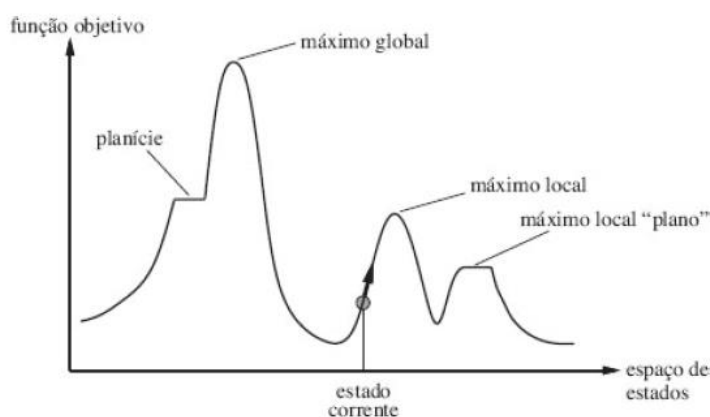
Na hibridização do AC com o BT, denominada AC+BT. O espaço de busca para o BT, é a vizinhança do melhor valor encontrado pelo AC a cada iteração. Desta forma tem-se sempre dados de entrada de qualidade na entrada da BT, proporcionando resultados melhores.

O parâmetro deixado como externo para manipulação do comportamento do AM definido por esta hibridização, foi o tamanho da lista tabu, que permite ajustar o ciclo de visita em uma considerada tabu. Este parâmetro é alterado pelo uso do arquivo XML de configurações, sem a necessidade de fazer intervenção no código fonte do *framework*.

### 4.3.2 Algoritmo Cultural com Hill Climbing (AC+HC)

O algoritmo memético gerado pela hibridização do AC com o HC, possui a variável “TamListaHC”, que é um vetor onde o seu tamanho define quantas vezes serão testados valores passados pelo AC. Este comprimento define o tamanho da “planície” a ser considerada no HC. Quando este valor é ultrapassado, um novo local na vizinhança será tomado para continuar a subida. A figura 4.5, mostra a função objetivo versus o espaço de estados.

**Figura 4.5:** Topologia de espaço de estados unidimensional.



**Fonte:** (NORVIG e RUSSELL, 2014).

Sendo que, cada busca será executada até que:

- Um número máximo de iterações definidas, seja realizado, ou;
- Não se encontrem melhoras significativas nos resultados obtidos.

É selecionado o melhor resultado encontrado.



### 4.3.3 Algoritmo Cultural com Simulated Annealing (AC+SA)

O recozimento simulado é uma metaheurística que possibilita o escape de mínimos locais, pelo controle do parâmetro temperatura. Nesta hibridização a variável que representa esse parâmetro é denominada de “Energia-BuscaLocal”. Este parâmetro é informado no arquivo XML de configurações.

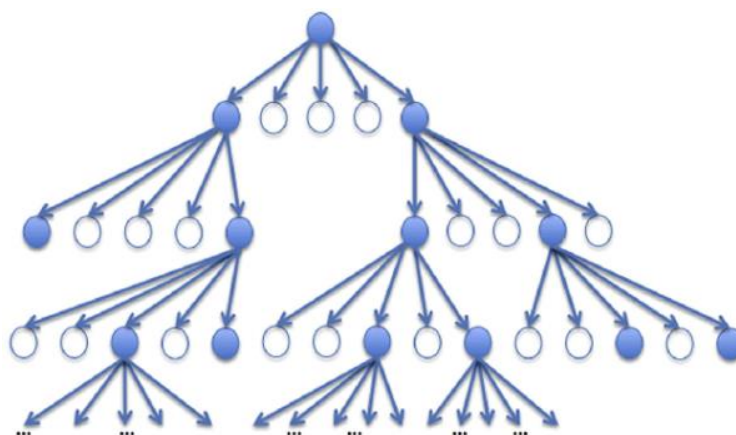
O SA trabalha de forma colaborativa com o AC. A cada iteração do AC é verificada a probabilidade de acionamento da busca local. Se esta for acionada, o indivíduo corrente tem sua vizinhança pesquisada pelo SA, que se move na sua estrutura de vizinhança definida por SILVA (2012b), na busca por melhores resultados.

### 4.3.4 Algoritmo Cultural com Beam Search (AC+BS)

A busca em feixe controla  $k$  estados. No início são gerados aleatoriamente  $k$  estados. Em cada passo são gerados  $k$  estados sucessores. O algoritmo irá parar se um destes for o objetivo. Senão, serão selecionados os  $k$  melhores sucessores a partir da lista completa e o procedimento será repetido.

No algoritmo memético deste trabalho que faz uso da hibridização do AC com o BS,  $k$  é representado pela variável “TamFeixesBS”. Os  $k$  estados gerados aleatoriamente, são substituídos por valores resultantes do AC. A figura 4.6, mostra um exemplo da estrutura da busca em feixes com  $k=2$ .

**Figura 4.6:** Estrutura da busca em feixes com  $k=2$ .



## 5 RESULTADOS

### 5.1 Introdução

A implementação do Algoritmo Memético Cultural com Busca Local – AMCBL, teve seu comportamento testado com funções padrões utilizados para benchmark de desempenho das técnicas evolutivas e tradicionais de otimização, híbridas ou não, e foi aplicado em soluções de otimização de problemas do mundo real nas áreas de engenharia elétrica, mecânica e civil.

### 5.2 Resultados para otimização de funções multimodais benchmark

#### 5.2.1 Resultados e discussões (Hellinger-TOPSIS)

Nas tabelas 5.1 e 5.2, são apresentados os resultados da formação da matriz de decisão com as médias e desvio padrão para cada alternativa (Algoritmos) ligados a cada critério (Funções), para o cenário 1, respectivamente.

**Tabela 5.1:** Matriz de Decisão das médias para o cenário 1.

Algoritmo	F1-10	F3-10	F5-10	F15-10
AC	3,23659E-10	0	7,79896E-10	0,058406595
AC+BS_4	3,41375E-10	0	6,8751E-11	0,051070190
AC+BS_8	3,98655E-10	0	3,54724E-10	0,043554249
AC+BS_12	2,99159E-10	0	3,10186E-09	0,046754072
AC+HC	2,03508E-10	0	2,29591E-09	0,047450157
AC+SA_5	2,42099E-10	0	1,11557E-09	0,052244734
AC+SA_10	4,08467E-10	0	3,0639E-10	0,051215125
AC+SA_15	3,15785E-10	0	1,06558E-10	0,054211218
AC+BT_2	1,92456E-10	0	1,13369E-09	0,043761561
AC+BT_4	4,21422E-10	0	8,171E-10	0,050767236
AC+BT_6	4,76319E-10	0	1,82737E-09	0,047881579

**Tabela 5.2:** Matriz de Decisão do desvio padrão para o cenário 1.

<b>Algoritmo</b>	<b>F1-10</b>	<b>F3-10</b>	<b>F5-10</b>	<b>F15-10</b>
AC	5,78075E-10	0	3,41E-09	0,066278
AC+BS_4	7,75806E-10	0	2,34E-10	0,035874
AC+BS_8	7,65625E-10	0	1,34E-09	0,025408
AC+BS_12	4,22803E-10	0	2,18E-08	0,021864
AC+HC	3,40307E-10	0	1,27E-08	0,031338
AC+SA_5	3,51351E-10	0	5,74E-09	0,029728
AC+SA_10	6,55284E-10	0	1,46E-09	0,029953
AC+SA_15	6,42544E-10	0	3,34E-10	0,033741
AC+BT_2	2,00562E-10	0	7,78E-09	0,026441
AC+BT_4	8,52425E-10	0	4,74E-09	0,026039
AC+BT_6	8,77602E-10	0	8,79E-09	0,033101

Os resultados da formação da matriz de decisão com as médias e desvio padrão para cada alternativa (Algoritmos) ligados a cada critério (Funções), para o cenário 2, são apresentadas nas tabelas 5.3 e 5.4, respectivamente.

**Tabela 5.3:** Matriz de Decisão das médias para o cenário 2.

<b>Algoritmo</b>	<b>F1-30</b>	<b>F3-30</b>	<b>F5-30</b>	<b>F15-30</b>
AC	4,13043E-06	0	1,237729462	0,021755013
AC+BS_4	8,53548E-06	0	2,54906941	0,007234179
AC+BS_8	1,24266E-05	0	1,027333483	0,010783016
AC+BS_12	5,43242E-06	0	0,5890824	0,006603329
AC+HC	8,67947E-06	0	0,993840572	0,011431659
AC+SA_5	9,97174E-06	0	1,193562951	0,009505563
AC+SA_10	7,8494E-06	0	6,571804288	0,030975012
AC+SA_15	6,14914E-06	0	2,592440713	0,005290837
AC+BT_2	4,39184E-06	0	2,444557802	0,006020829
AC+BT_4	4,78786E-06	0	0,023245916	0,003247642
AC+BT_6	4,69211E-06	0	1,493836074	0,005106656

**Tabela 5.4:** Matriz de Decisão do desvio padrão para o cenário 2.

Algoritmo	F1-30	F3-30	F5-30	F15-30
AC	4,6564E-06	0	4,327119234	0,106551678
AC+BS_4	1,6813E-05	0	11,50307561	0,01226867
AC+BS_8	3,90693E-05	0	4,503880093	0,031645423
AC+BS_12	7,49513E-06	0	2,868408437	0,012005268
AC+HC	1,5575E-05	0	5,164115945	0,026760044
AC+SA_5	2,05916E-05	0	6,281450981	0,018866872
AC+SA_10	1,13703E-05	0	29,52707128	0,123691902
AC+SA_15	7,39825E-06	0	12,99087696	0,01235467
AC+BT_2	5,1452E-06	0	9,044781792	0,010675999
AC+BT_4	6,0466E-06	0	0,074867226	0,006271337
AC+BT_6	5,1039E-06	0	6,993637807	0,009563504

Com as matrizes de decisão compostas com os valores da média e do desvio padrão obtidos após cada simulação dos algoritmos em cada uma das funções, foram montadas as tabelas 5.5 e 5.6 com os valores do PIS e NIS para cada critério do cenário 1.

**Tabela 5.5:** PIS e NIS para as médias de cada critério da Matriz de Decisão para o cenário 1.

Critérios	F1-10	F3-10	F5-10	F15-10
PIS	1,9246E-10	0	6,8751E-11	0,043554249
NIS	4,7632E-10	0	3,10186E-09	0,058406595

**Tabela 5.6:** PIS e NIS para o Desvio Padrão cada critério da Matriz de Decisão para o cenário 1.

Critérios	F1-10	F3-10	F5-10	F15-10
PIS	2,0056E-10	0	2,34023E-10	0,021864114
NIS	8,7760E-10	0	2,17959E-08	0,066278482

As tabelas 5.7 e 5.8, mostram os valores do PIS e NIS para cada critério do cenário 2.

**Tabela 5.7:** PIS e NIS para as médias de cada critério da Matriz de Decisão para o cenário 2.

Critérios	F1-30	F3-30	F5-30	F15-30
PIS	4,1304E-06	0	0,023245916	0,003247642
NIS	1,2427E-05	0	6,571804288	0,030975012

**Tabela 5.8:** PIS e NIS para o Desvio Padrão cada critério da Matriz de Decisão para o cenário 2.

Critérios	F1-30	F3-30	F5-30	F15-30
PIS	4,6564E-06	0	0,074867226	0,006271337
NIS	3,9069E-05	0	29,52707128	0,123691902

A tabulação dos valores dos PIS e do NIS para cada critério servem de base para o cálculo da medida de separação  $d^+$  do PIS ( $f^+$ ) e para o  $d^-$  do NIS ( $f^-$ ). Com estas medidas é possível calcular o coeficiente de proximidade relativa  $\xi_i$  para cada algoritmo com relação ao PIS. O maior valor de  $\xi_i$ , representa uma melhor colocação para os algoritmos aos resultados obtidos para as funções de testes. As tabelas 5.9 e 5.10, mostram os valores das medidas de separação e do coeficiente de proximidade, dispostos em ranks que mostram a classificação de cada algoritmo no cenário 1 e 2, respectivamente.

**Tabela 5.9:** Resultado para o cenário 1.

Algoritmo	$d^+$	$d^-$	$\xi_i$	Rank
AC+BS_4	1,8111	2,3036	0,5599	1
AC+SA_15	1,4585	1,7697	0,5482	2
AC+BS_8	2,2846	2,3201	0,5039	3
AC+BT_4	2,3124	2,3447	0,5035	4
AC+BT_6	2,0376	1,9181	0,4849	5
AC+BT_2	2,1216	1,9740	0,4820	6
AC+HC	2,3379	2,0471	0,4668	7
AC+BS_12	2,3383	1,8344	0,4396	8
AC+SA_5	2,0832	1,6044	0,4351	9
AC+SA_10	1,3581	1,0309	0,4315	10
AC	2,7547	1,8859	0,4064	11

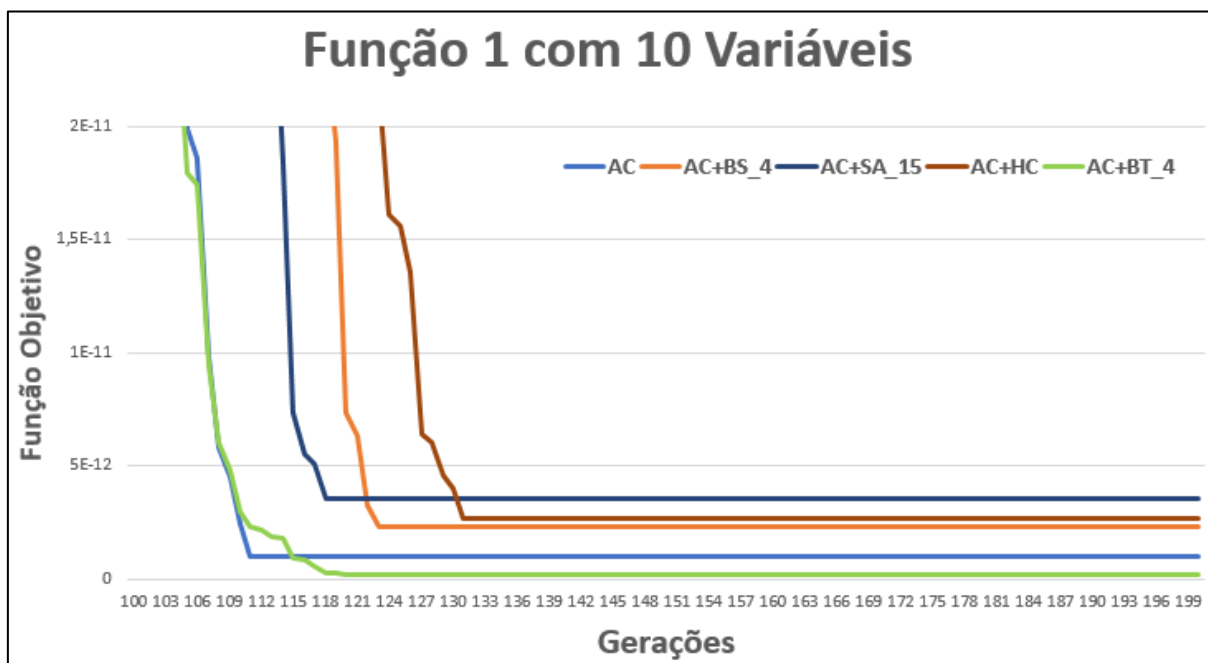
**Tabela 5.10:** Resultado para o cenário 2.

Algoritmo	$d^+$	$d^-$	$\xi_i$	Rank
AC+BS_12	2,4495	3,1338	0,5613	1
AC+BT_6	2,4155	2,2826	0,4859	2
AC	2,7167	2,4855	0,4778	3
AC+BS_4	2,8283	2,5609	0,4752	4
AC+SA_5	2,5017	2,2447	0,4729	5
AC+HC	3,0247	2,6596	0,4679	6
AC+SA_15	2,1870	1,8987	0,4647	7
AC+BT_4	3,0125	2,5866	0,4620	8
AC+BT_2	2,5174	1,9099	0,4314	9
AC+SA_10	2,0987	1,5487	0,4246	10
AC+BS_8	3,2522	2,2352	0,4073	11

A figura 5.1, mostra o gráfico da função 1 (F1) com 10 variáveis para os algoritmos de melhor classificação: AC, AC+BS\_4, AC+SA\_15, AC+HC e AC+BT\_4 no cenário1. Ou seja, o AC+BS\_4 (Tamanho do feixe igual a 4), o AC+SA\_15 (Energia de busca local igual a 15), AC+BT\_4 (comprimento da lista tabu igual a 4), AC e o AC+HC em sua configuração

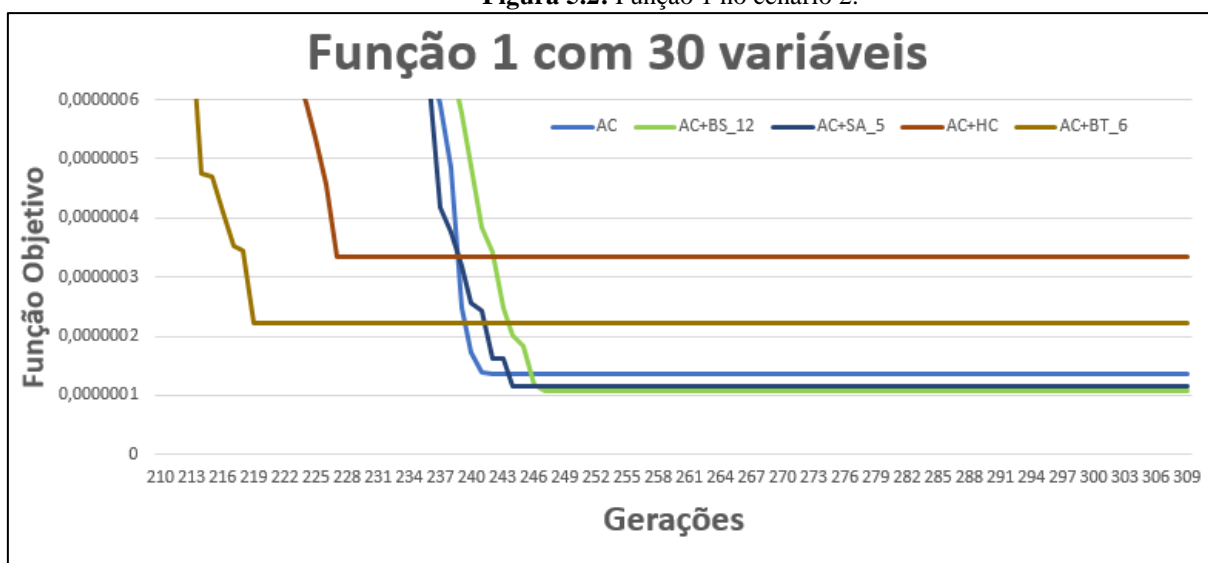
básica. Nesta figura se nota que o AC+BT\_4 obteve uma convergência maior para o resultado esperado. Já, o AC foi o segundo melhor colocado e os demais, neste cenário, ficaram mais distante do objetivo.

**Figura 5.1:** Função 1 no cenário 1.



A figura 5.2, mostra o gráfico da função 1 (F1) com 30 variáveis para os algoritmos de melhor classificação: AC, AC+BS\_12, AC+SA\_5, AC+HC e AC+BT\_6 no cenário2 que foi evoluído em 600 gerações. Para este cenário tivemos o AC+BS\_12 (Tamanho do feixe igual a 12), com o melhor resultado. O AC+SA\_5 (Energia de busca local igual a 5) ficou em segundo lugar. Neste cenário o AC ainda superou o AC+HC e o AC+BT\_6 (comprimento da lista tabu igual a 6).

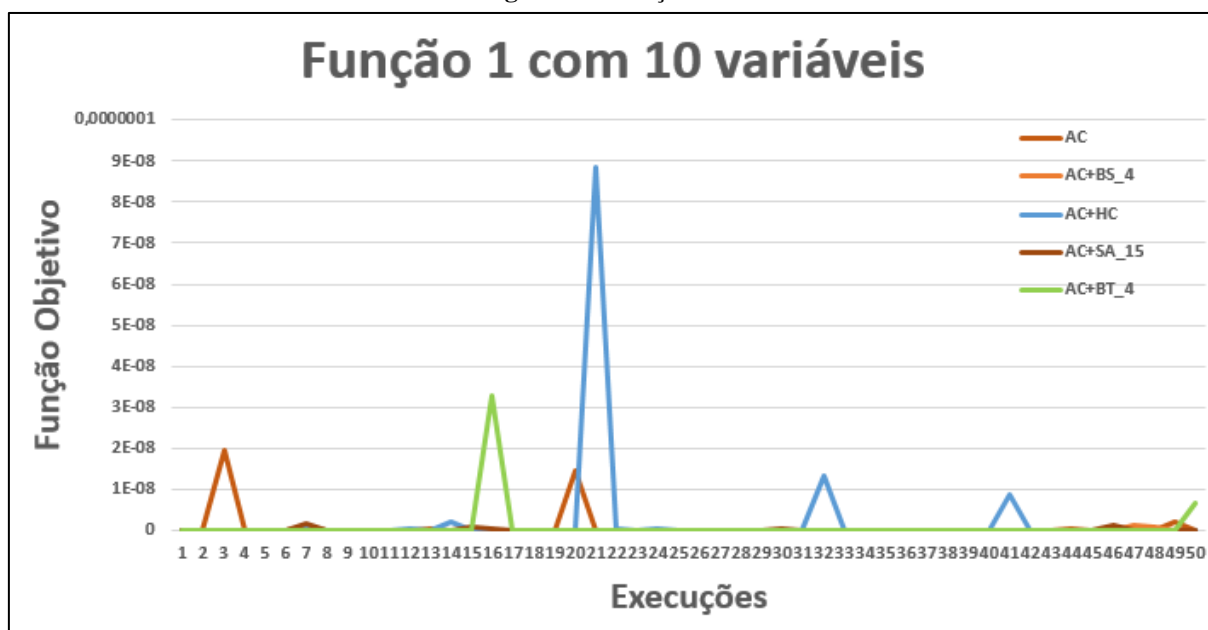
Figura 5.2: Função 1 no cenário 2.



As figuras 5.3 e 5.4 apresentam os gráficos da função 1 (F1) com os algoritmos de melhor classificação em cada cenário. Nestas figuras é mostrado a evolução do fitness em função de cada iteração.

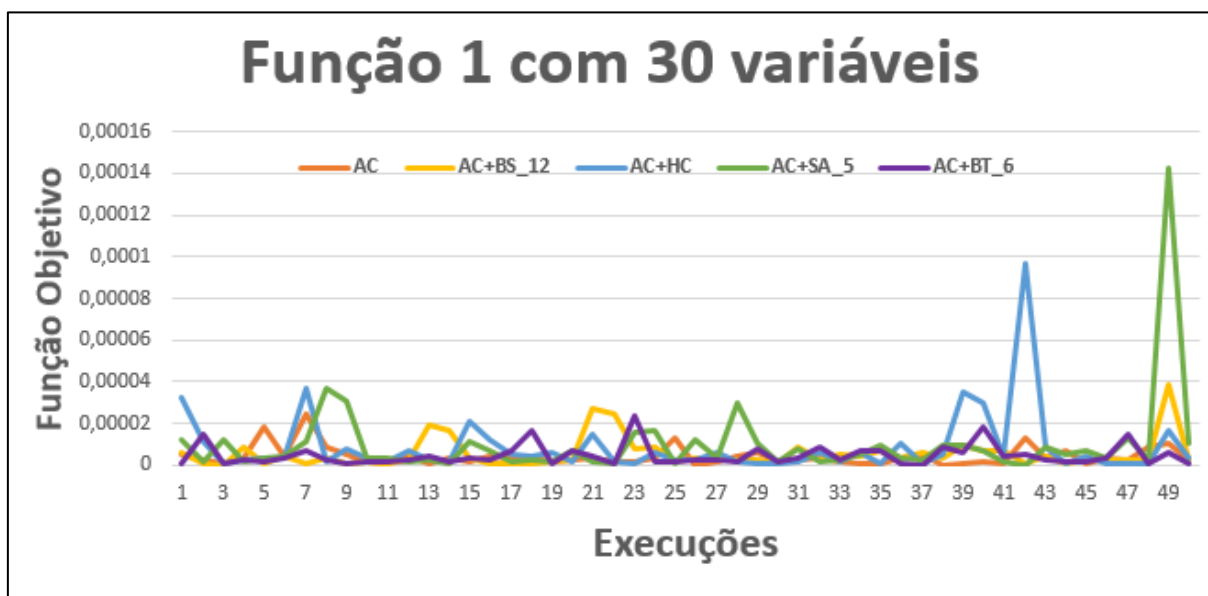
Os valores finais para os cinco algoritmos (AC, AC+BS\_4, AC+HC, AC+SA\_15 e AC+BT\_4) variam em uma faixa menor que  $3,5E-08$ , somente o AC+HC em uma execução alcançou valor fora desta faixa (Figura 5.3).

Figura 5.3: Função 1 no cenário 1.



A figura 5.4, mostra que os valores finais para os cinco algoritmos variam em uma faixa menor que 0,00004. O AC+HC e o AC+SA\_5 tiveram um ponto fora desta faixa.

**Figura 5.4:** Função 1 no cenário 2.



Na comparação das tabelas 5.9 e 5.10 das simulações realizadas, nota-se que os algoritmos híbridos sempre estiveram nas duas primeiras posições no *rank* em ambos cenários. O AC ocupou a posição mais baixa no *rank* quando aplicado no cenário 1. Para o cenário 2, o AC ficou em terceiro lugar para as funções testadas. As Figuras 5.1 e 5.2 foram elaboradas com a função 1 selecionada aleatoriamente, utilizando a iteração com os melhores resultados para observar o comportamento das gerações. Mostrando que no cenário 1, a partir da 110ª geração o AC encontrou o seu valor final e não houve melhoras em seu resultado. Nos demais algoritmos com busca local obtiveram melhoras no decorrer da evolução das gerações. Para este cenário o AC+BT\_4 obteve melhor resultado. Para o cenário 2, o AC+BS\_12 foi o melhor a partir da 247ª geração. As Figuras 5.3 e 5.4, também são da função 1. Porém, estes gráficos foram elaborados com o resultado final de cada execução, isto para se observar se existe uma uniformidade nos valores encontrados e assinalados como os melhores resultados de cada execução. Para estes dois últimos gráficos pode-se observar que no cenário 1, o AC+HC foi o único que teve um valor muito maior que os demais na iteração 21. No cenário 2, o AC+HC e o AC+SA\_5 tiveram picos maiores. Os demais se mantiveram em uma faixa com valores muito próximos.



### 5.2.2 Resultados e discussões (Friedman, Friedman Aligned e Quade)

Para avaliação, os resultados dos melhores valores de cada simulação foram submetidos aos testes de Friedman, Friedman Aligned e Quade. Os testes foram realizados com os resultados de D=10, D=30 e com a junção destes dois resultados. As entradas de dados para análise estão nas tabelas 5.11 e 5.12.

**Tabela 5.11:** Entrada de dados de D-10.

Data-set	F1-10	F3-10	F5-10	F15-10	FH1-10	FH4-10	FH5-10	FH8-10
AC	2,47E-12	0	5,00E-15	0,004932	7,75E-03	1,069561	4,37E-01	1,569561
AC+BS_4	3,27E-12	0	7,10E-14	9,39E-12	1,02E-05	1,069561	5,07E-01	1,569561
AC+BS_8	2,16E-13	0	2,50E-14	0,004932	8,50E-03	1,069561	4,17E-01	1,569561
AC+BS_12	3,88E-12	0	1,10E-14	0,004932	1,50E-02	1,069561	3,97E-01	1,569561
AC+HC	2,66E-12	0	4,00E-15	0,014772	1,09E-01	1,069561	6,58E-01	1,569561
AC+SA_5	4,74E-12	0	1,60E-14	2,71E-11	2,26E-06	1,069561	4,63E-01	1,569561
AC+SA_10	2,84E-12	0	0,00E+00	4,46E-12	1,11E-06	1,069561	2,80E-01	1,569561
AC+SA_15	5,07E-12	0	5,00E-15	0,009857	1,48E-03	1,069561	5,04E-01	1,569561
AC+BT_2	4,98E-12	0	0,00E+00	2,74E-12	2,43E-02	1,069561	3,78E-01	1,569561
AC+BT_4	2,43E-13	0	3,00E-14	1,64E-12	1,82E-03	1,069561	5,59E-01	1,569561
AC+BT_6	5,07E-12	0	5,00E-15	0,004932	1,44E-02	1,069561	3,91E-01	1,569561

**Tabela 5.12:** Entrada de dados de D-30.

Data-set	F1-30	F3-30	F5-30	F15-30	FH1-30	FH4-30	FH5-30	FH8-30
AC	1,36E-07	0	1,48E-08	5,86E-06	1,37E+01	1,075758	2,07E+00	1,569607
AC+BS_4	3,91E-07	0	1,24E-09	1,12E-05	1,37E+01	1,07522	1,54E+00	1,569598
AC+BS_8	2,41E-07	0	1,02E-08	8,26E-06	1,37E+01	1,074143	1,74E+00	1,569685
AC+BS_12	1,07E-07	0	9,66E-10	1,24E-05	1,37E+01	1,074362	8,89E-01	1,569618
AC+HC	4,57E-07	0	4,72E-09	6,03E-06	1,37E+01	1,075159	1,52E+00	1,569614
AC+SA_5	1,61E-07	0	6,31E-09	5,79E-06	1,37E+01	1,07388	8,98E-01	1,569669
AC+SA_10	6,08E-07	0	1,87E-08	6,96E-06	1,37E+01	1,077677	3,77E+00	1,569638
AC+SA_15	4,17E-08	0	7,48E-09	1,22E-05	1,37E+01	1,074382	7,13E+00	1,569601
AC+BT_2	2,13E-07	0	1,30E-08	9,9E-06	1,37E+01	1,075381	8,51E-01	1,569638
AC+BT_4	8,97E-08	0	1,59E-08	8,19E-06	1,37E+01	1,07202	2,17E+00	1,569655
AC+BT_6	2,22E-07	0	4,53E-09	8,38E-06	1,37E+01	1,074508	3,07E+00	1,56961

Os dados das tabelas 5.11 e 5.12 foram submetidos a avaliação dos testes tratados neste trabalho. Obteve-se como resultados os dados mostrados na tabela 5.13, 5.14 e 5.15. Os valores em vermelho são os melhores resultados em cada teste associado ao algoritmo que teve melhor desempenho para o conjunto de funções. Os valores em azul são os que ocuparam o segundo lugar no *rank* e os valores em verde são os que ficaram em terceiro lugar.

Tabela 5.13: Resultado para D-10.

Data-set	Friedman	Friedman Aligned	Quade
AC	5,9375	51,5625	5,6667
AC+BS_4	6,0625	42,4375	6,2778
AC+BS_8	5,8125	43,3125	5,8056
AC+BS_12	4,8750	44,0000	5,1944
AC+HC	4,4375	31,5625	3,1667
AC+SA_5	6,6250	43,2500	6,9861
AC+SA_10	8,9375	61,6875	9,7222
AC+SA_15	5,0625	37,0625	4,8056
AC+BT_2	7,0000	47,5000	7,1250
AC+BT_4	5,6875	40,6875	5,5833
AC+BT_6	5,5625	46,4375	5,6667

Tabela 5.14: Resultado para D-30.

Data-set	Friedman	Friedman Aligned	Quade
AC	5,8750	42,0000	5,5833
AC+BS_4	6,1250	44,3750	6,0278
AC+BS_8	5,3750	50,2500	5,5833
AC+BS_12	6,7500	47,2500	6,6111
AC+HC	6,8750	50,0000	7,3056
AC+SA_5	7,6250	55,1250	8,0556
AC+SA_10	4,3750	30,8750	4,6389
AC+SA_15	6,3750	45,5000	5,9722
AC+BT_2	5,6250	40,3750	6,1944
AC+BT_4	5,5000	43,2500	5,2222
AC+BT_6	5,5000	40,5000	4,8056

Sendo, a tabela 5.13, o resultado obtido quando se aplicou somente o conjunto de dados com dimensão 10. Nesta tivemos o AC+HC como algoritmo melhor classificado nos três testes. O AC+SA\_15 foi classificado pelo teste Friedman em terceiro lugar e pelos outros dois testes em segundo lugar. Ainda foram classificados o AC+BS\_12 pelos testes de Friedman e Quade, em segundo e terceiro lugar respectivamente. O AC+BT\_4 foi classificado pelo teste Friedman Aligned em terceiro lugar no *rank*.

A tabela 5.14 mostra os valores resultantes dos testes para o conjunto de dados com dimensão 30 em cada problema. Sendo neste cenário o AC+SA\_10 o primeiro colocado no rank de todos os testes. Sendo também classificados em segundo e terceiro lugares os algoritmos: AC+BT\_6, AC+BT\_4, AC+BS\_8 e AC+BT\_2.

Após avaliação individual dos algoritmos propostos com os devidos problemas, resolveu-se juntar as bases de dados geradas com dimensões 10 e 30, para verificar se a classificação destes três testes se diferencia muito dos resultados anteriores. A tabela 5.15 mostras o resultado após a submissão destes dados aos testes aqui abordados. Para o resultado da união dos valores obtidos notou-se que este acompanhou a simulação para dimensão 10 para os testes de Friedman Aligned e Quade. Observou-se ainda que, os seis algoritmos que foram classificados neste cenário estavam presentes entre os primeiros colocados nos dois cenários de testes realizados anteriormente.

**Tabela 5.15:** Resultado para a junção de D-10 e D-30.

Data-set	Friedman	Friedman Aligned	Quade
AC	5,9063	91,6563	5,6140
AC+BS_4	6,0938	86,8438	6,2096
AC+BS_8	5,5938	94,2813	5,6434
AC+BS_12	5,8125	92,4375	6,0515
AC+HC	5,6563	80,0313	5,2096
AC+SA_5	7,1250	99,9375	7,5772
AC+SA_10	6,6563	87,2188	6,8199
AC+SA_15	5,7188	82,4063	5,4890
AC+BT_2	6,3125	84,8750	6,5625
AC+BT_4	5,5938	85,0313	5,4890
AC+BT_6	5,5313	88,7813	5,3346

### 5.3 Aplicação na solução de problemas de otimização em engenharia

#### 5.3.1 Análise dos resultados (Engenharia elétrica)

##### A. RESULTADOS DA PLANTA REAL DA GERAÇÃO DE ENERGIA

Após 30 simulações, cada simulação com uma população inicial diferente, verificou-se que os valores obtidos para o custo e emissões foram melhores que os obtidos nos artigos utilizados como referência (NASCIMENTO *et al.*, 2016; JÚNIOR *et al.*, 2017). A Tabela 5.16, mostra os valores encontrados por (NASCIMENTO *et al.*, 2016; JÚNIOR *et al.*, 2017), que resolveram o problema de CEELD com ED (Evolução Diferencial) e com SA, respectivamente, em comparação com o AC clássico deste trabalho. O resultado em azul mostra o melhor resultado dos algoritmos. O resultado verde mostra o melhor resultado do algoritmo quando os geradores com o maior custo operacional estão desligados.

**Tabela 5.16:** Relatório final do comparativo multi-objetivo AC e SA (Energia gerada por gerador).

CEELD usando SA e AC como solução:	ED Clássico (NASCIMENTO <i>et al.</i> , 2016)	ED Modificado Desligando geradores (NASCIMENTO <i>et al.</i> , 2016)	SA Clássico (JÚNIOR <i>et al.</i> , 2017)	SA Modificado Desligando geradores (JÚNIOR <i>et al.</i> , 2017)	AC Clássico	AC Modificado Desligando geradores
Demanda:	20 MW	20 MW	20 MW	20 MW	20 MW	20 MW
Potência Mínima:	0.56 MW	0.56 MW	0.56 MW	0.56 MW	0.56 MW	2.10 MW
Potência Máxima:	3.9 MW	3.7 MW	3.9 MW	3.7 MW	3.9 MW	3.9 MW
Custo do Combustível:	<b>1922.72 \$/h</b>	1544.04 \$/h	1962.45 \$/h	1548.13 \$/h	1959.42 \$/h	<b>1292.60 \$/h</b>
Total de Emissões	-	-	2744.41 kg/h	1754.74 kg/h	<b>1354.19 kg/h</b>	<b>1658.74 kg/h</b>
Potência perdida:	0.01	0.01	0.01	0.01	0.01	0.01
Potência de cada gerador em MW:						
Pm1	3.35	3.35	1.34	3.27	1.09	Desligado
Pm2	3.70	3.70	1.04	2.56	3.70	3.70
Pm3	3.60	3.46	2.24	3.54	2.00	Desligado
Pm4	2.16	2.97	2.25	3.33	1.98	2.73
Pm5	3.45	3.08	3.19	3.33	1.86	Desligado
Pm6	0.66	Desligado	2.93	Desligado	1.84	2.52
Pm7	0.88	Desligado	2.10	Desligado	1.82	2.51
Pm8	0.75	2.92	2.06	2.13	1.86	2.54
Pm9	0.90	Desligado	2.21	Desligado	3.90	3.90
Pm10	0.56	0.56	1.81	1.85	0.56	2.11
Total Potência	20.01	20.03	20.50	20.01	20.61	20.00

A comparação feita na Tabela 5.16, mostra que os resultados usando o AC foram melhores que os obtidos por (JÚNIOR *et al.*, 2017), com a aplicação do SA, tanto para custo quanto para emissões. No entanto, o melhor custo foi encontrado em (NASCIMENTO *et al.*, 2016), onde ED foi usado, porém, vale ressaltar que o ED foi aplicado somente na função de custo, ou seja, não foi uma aplicação multiobjetivo. Na avaliação da aplicação das técnicas listadas na Tabela 5.16, aplicada em todos os geradores, houve uma redução de 50,66% das emissões com o uso de AC. Quando foi utilizado o recurso para desligar os geradores com maior custo na geração de energia, os valores de redução com o uso da AC foram: 16,28% para o custo do combustível e 5,47% para as emissões.

Como o uso do AC implementado neste trabalho obteve valores melhores que o trabalho de (JÚNIOR *et al.*, 2017), o restante do trabalho irá comparar os resultados do AC com a hibridização do AC com o SA e depois com o BT.

A Tabela 5.17, mostra o relatório final com a demonstração dos melhores valores de  $P_i$  encontrados por cada algoritmo. Observa-se que os valores são muito próximos em cada cenário (cenário 1: uso de todos os geradores, cenário 2: possibilidade de desligar os geradores com maior custo na geração de energia). No entanto, há uma pequena melhoria nos resultados obtidos com o AC híbrido com as pesquisas locais. No entanto, o melhor resultado foi o AC+BT que usou a Lista Tabu com tamanho igual a 6.

**Tabela 5.17:** Comparativo multi-objetivo AC com Híbrido (Energia gerada por gerador).

CEELD usando AC como solução:	AC Clássico	AC Modificado Desligando geradores	AC+SA	AC+SA Modificado Desligando geradores	AC+BT	AC+BT Modificado Desligando geradores
Demanda:	20 MW	20 MW	20 MW	20 MW	20 MW	20 MW
Potência Mínima:	0.56 MW	2.10 MW	0.56 MW	0.56 MW	0.56 MW	2.10 MW
Potência Máxima:	3.9 MW	3.9 MW	3.9 MW	3.9 MW	3.9 MW	3.9 MW
Custo do Combustível:	1959.42 \$/h	1292.60 \$/h	1959.52 \$/h	1292.40 \$/h	<b>1958.62 \$/h</b>	<b>1292.37 \$/h</b>
Total de Emissões	1354.19 kg/h	1658.74 kg/h	1356.56 kg/h	1656.71 kg/h	<b>1345.89 kg/h</b>	<b>1656.70 kg/h</b>
Potência perdida:	0.01	0.01	0.01	0.01	0.01	0.01
Potência de cada gerador em MW:						
Pm1	1.09	Desligado	1.07	Desligado	1.09	Desligado
Pm2	3.70	3.70	3.70	3.70	3.68	3.70
Pm3	2.00	Desligado	2.01	Desligado	2.01	Desligado
Pm4	1.98	2.73	2.01	2.73	1.98	2.73
Pm5	1.86	Desligado	1.87	Desligado	1.86	Desligado
Pm6	1.84	2.52	1.86	2.55	1.83	2.56
Pm7	1.82	2.51	1.78	2.47	1.81	2.43
Pm8	1.86	2.54	1.86	2.54	1.83	2.56
Pm9	3.90	3.90	3.90	3.90	3.90	3.90
Pm10	0.56	2.11	0.56	2.11	0.56	2.12
Total Potência	20.61	20.00	20.62	20.00	20.55	20.00

A Tabela 5.18, mostra os valores de emissão de cada gerador, que são obtidos quando aplicados (2), usando os valores de  $P_i$  encontrados por cada algoritmo. Observa-se que as emissões totais aumentam quando se utiliza a característica de desligar os geradores de maior custo na geração de energia. Isso se deve ao fato de que, nesse cenário, a prioridade é reduzir o custo do combustível.

**Tabela 5.18:** Comparativo multi-objetivo AC com Híbrido (Emissão por gerador).

CEELD usando AC como solução:	AC Clássico (m <sup>3</sup> /h)	AC Modificado Desligando geradores (m <sup>3</sup> /h)	AC+SA (m <sup>3</sup> /h)	AC+SA Modificado Desligando geradores (m <sup>3</sup> /h)	AC+BT (m <sup>3</sup> /h)	AC+BT Modificado Desligando geradores (m <sup>3</sup> /h)
Em1	89.20	Desligado	85.20	Desligado	89.20	Desligado
Em2	190.95	190.95	190.94	190.95	188.90	190.95
Em3	159.98	Desligado	161.91	Desligado	161.59	Desligado
Em4	156.79	298.62	161.95	298.62	156.79	298.62
Em5	147.46	Desligado	149.28	Desligado	147.46	Desligado
Em6	144.29	271.12	148.06	277.62	142.72	279.81
Em7	141.16	268.97	135.53	260.44	139.61	252.06
Em8	147.46	275.45	146.79	275.45	142.72	279.82
Em9	163.73	163.73	163.73	163.73	163.73	163.73
Em10	13.17	189.90	13.17	189.90	13.17	191.71
Total de Emissões	1354.19	1658.74	1356.56	1656.71	<b>1345.89</b>	<b>1656.70</b>

A Tabela 5.19, mostra os custos de combustível para cada gerador, que são obtidos ao aplicar (1), usando os valores de  $P_i$  encontrados por cada algoritmo. Observa-se que houve redução de aproximadamente 34,00% no custo total de combustível nos cenários utilizados

(cenário 1: uso de todos os geradores, cenário 2: possibilidade de desligar os geradores com maior custo de geração de energia). Outro ponto a ser destacado, é que o AC com o BT, apresentou uma pequena melhora nos valores obtidos em relação aos demais algoritmos

**Tabela 5.19:** Comparativo multi-objetivo AC com Híbrido (Custo por gerador).

CEELD usando AC como solução:	AC Clássico (USD/h)	AC Modificado Desligando geradores (USD/h)	AC+SA (USD/h)	AC+SA Modificado Desligando geradores (USD/h)	AC+BT (USD/h)	AC+BT Modificado Desligando Geradores (USD/h)
Cost1	247.64	Desligado	247.46	Desligado	247.64	Desligado
Cost2	237.13	237.13	237.13	237.13	236.93	237.13
Cost3	237.04	Desligado	237.14	Desligado	237.12	Desligado
Cost4	221.82	230.10	222.17	230.10	221.82	230.10
Cost5	239.56	Desligado	239.68	Desligado	239.56	Desligado
Cost6	142.11	150.29	142.39	150.65	141.99	150.77
Cost7	155.50	165.19	154.99	164.63	155.36	164.06
Cost8	156.07	165.61	156.01	165.61	155.65	165.89
Cost9	184.71	184.71	184.71	184.71	184.71	184.71
Cost10	137.84	159.57	137.84	159.57	137.84	159.71
Custo Total	1959.42	1292.60	1959.52	1292.40	<b>1958.62</b>	<b>1292.37</b>

Os resultados do algoritmo de otimização computacional baseado em algoritmos culturais mostram que houve uma minimização das variáveis do custo total de combustível e das emissões de gases, mantendo a demanda programada. A Tabela 5.20, compara os valores de trabalho de (JÚNIOR *et al.*, 2017) com cada algoritmo do trabalho apresentado neste artigo. Observa-se que os melhores resultados foram obtidos com a hibridação com a busca local. Isso se deve ao fato de que a AC encontra o valor ideal por sua exploração e a pesquisa local realiza uma nova exploração na vizinhança do ponto encontrado pela AC. Os resultados das pesquisas locais foram muito próximos, mas, AC+BT, obteve melhores valores.

**Tabela 5.20:** SA Clássico *versus* Algoritmos deste trabalho.

SA Clássico vs Algoritmos deste trabalho					
Algoritmo	Custo Total USD/h	% do custo em relação ao SA	Total Emissões m <sup>3</sup> /h	% de emissões em relação ao SA	
SA Clássico (JÚNIOR <i>et al.</i> , 2017)	1962.45	-	2744.41	-	
AC Clássico	1959.42	- 0.15%	1354.19	- 50.66%	
AC+SA	1959.52	- 0.15%	1356.56	- 50.57%	
AC+BT	<b>1958.62</b>	- 0.20%	<b>1345.89</b>	- 50.96%	
SA Clássico Desligando geradores (JÚNIOR <i>et al.</i> , 2017)	1548.13	-	1754.74	-	
AC Modificado Desligando geradores	1292.60	- 16.51%	1658.74	- 5.47%	
AC+SA Modificado Desligando geradores	1292.40	- 16.52%	1656.71	- 5.59%	
AC+BT Modificado Desligando geradores	<b>1292.37</b>	- 16.52%	<b>1656.70</b>	- 5.59%	

A Tabela 5.21 compara os valores de custo de combustível de (NASCIMENTO *et al.*, 2016) com cada algoritmo apresentado neste trabalho.

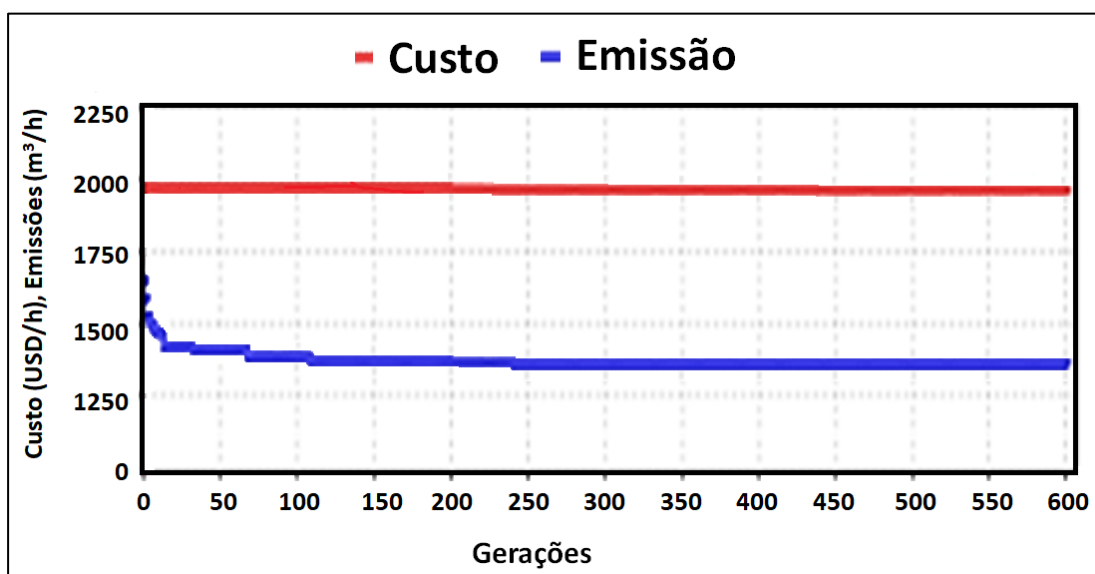
**Tabela 5.21:** DE Clássico *versus* Algoritmo deste trabalho.

DE Clássico vs Algoritmos deste trabalho		
Algoritmo	Custo Total USD/h	% do custo em relação ao DE
DE Clássico (NASCIMENTO <i>et al.</i> , 2016)	1922.72	-
AC Clássico	1959.42	1,91%
AC+SA	1959.52	1,91%
AC+BT	1958.62	1,87%
DE Clássico Desligando geradores (NASCIMENTO <i>et al.</i> , 2016)	1544.04	-
AC Modificado Desligando geradores	1292.60	- 16,28%
AC+SA Modificado Desligando geradores	1292.40	- 16,30%
AC+BT Modificado Desligando geradores	1292.37	- 16,30%

Como o AC+BT obteve melhores resultados que os demais algoritmos, foi utilizado para gerar as curvas das figuras 5.5, 5.6, 5.7 e 5.8.

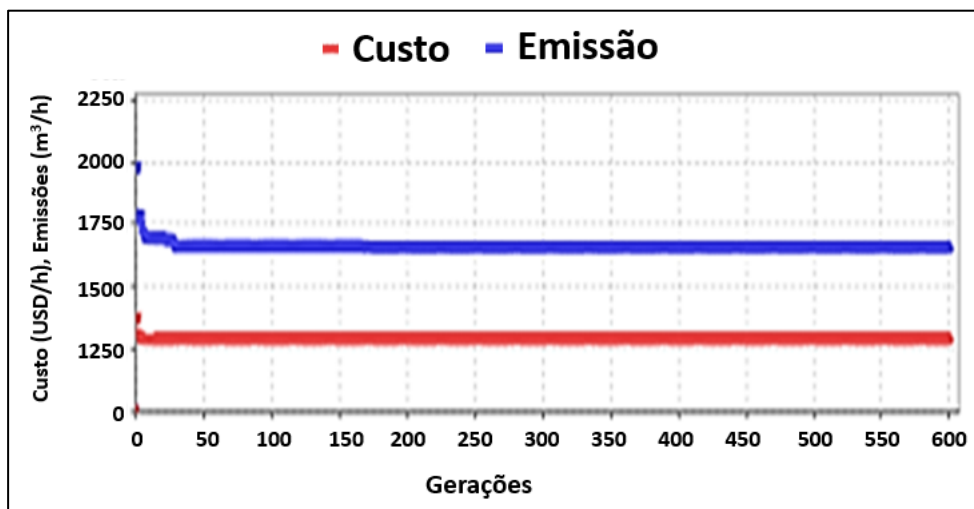
A figura 5.5 e a figura 5.6, mostram a evolução das funções de custo e emissão em relação às gerações, na situação de uso de todos os geradores e com desconexões de geradores com maior custo incremental, respectivamente.

**Figura 5.5:** Função de Custos e função de emissões vs Gerações, usando todos os motores.



Na Figura 5.5, observa-se que quando todos os geradores são utilizados, a função de emissão diminui seu valor até a geração 250 e então se torna um valor constante.

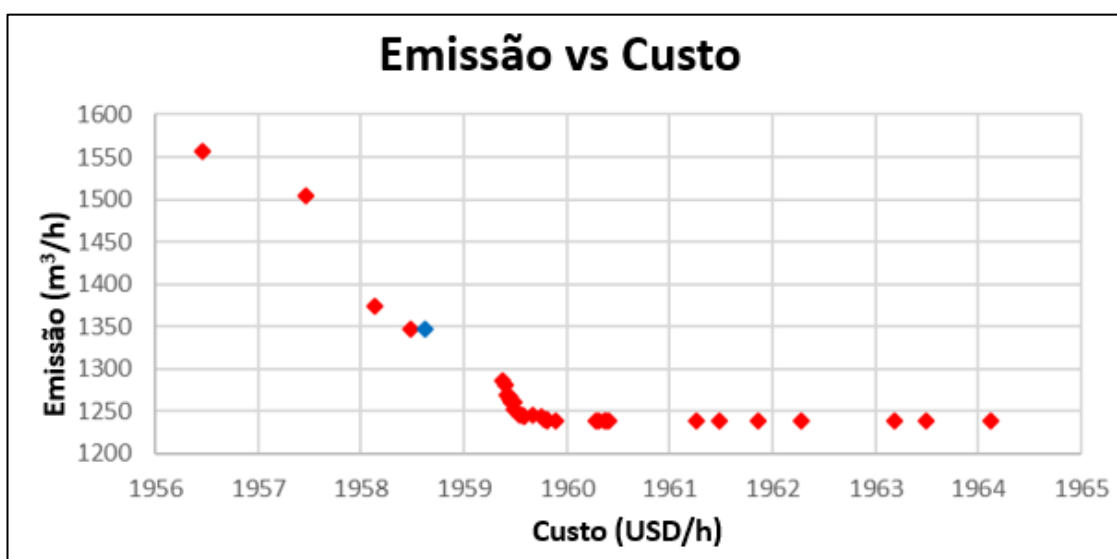
**Figura 5.6:** Função de Custos e função de emissões vs Gerações, desligando os motores com maior custo incremental.



Na Figura 5.6, uma vez que os geradores que apresentam maior custo na geração de energia estão desligados, a função custo já apresenta um valor muito inferior ao apresentado na Fig. 5.5. Entretanto, a função de emissão possui um valor maior que o apresentado na situação anterior.

A Figura 5.7 e a Figura 5.8, mostram a frente de Pareto das funções custo versus emissão, na situação de uso de todos os geradores e com desligamentos de geradores com o maior custo incremental, respectivamente.

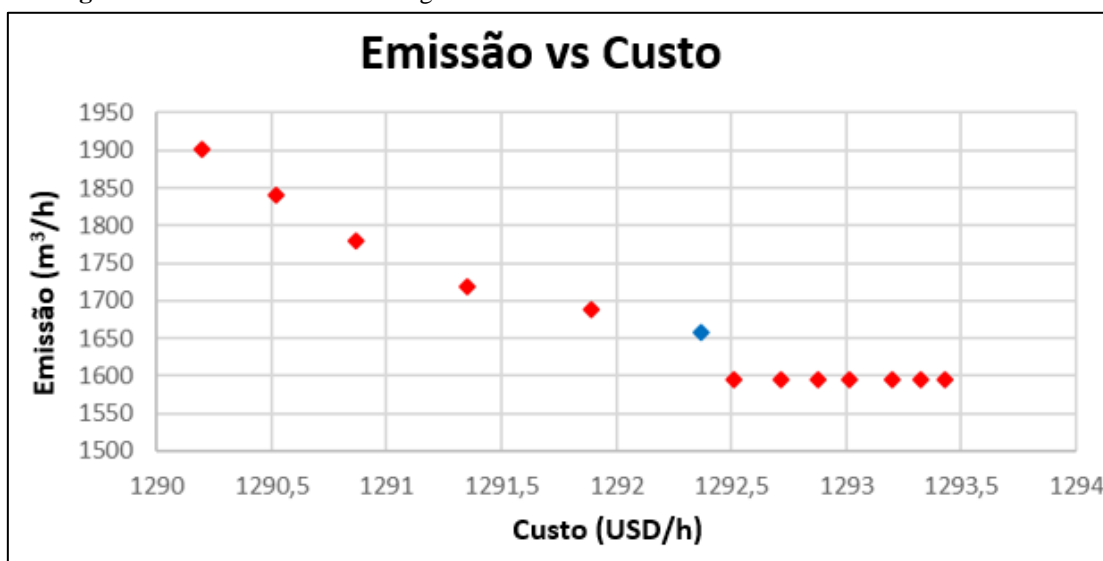
**Figura 5.7:** Frente de Pareto dos custos versus emissões usando todos os motores.





Para todos os pontos da curva de Frente de Pareto, a energia total gerada é igual à demanda de 20 MW. No entanto, no gráfico mostrado na Figura 5.7, observa-se que, se o gerente da usina térmica quiser reduzir bastante o custo, terá um aumento significativo de emissões. Por exemplo, custos inferiores a 1965,00 USD / h, fazem com que os valores de emissão sejam superiores a 1450,00 m<sup>3</sup> / h. O inverso ocorre para emissões inferiores a 1360,00 m<sup>3</sup> / h, os custos são superiores a 1967,50 USD / h.

**Figura 5.8:** Frente de Pareto desligando os motores com maior custo incremental.



O gráfico mostrado na Figura 5.8 tem um comportamento de emissão mais crítico. Para pequenas variações no custo do combustível, há uma grande variação nas emissões. Pois nessa abordagem, com geradores desligados pelo alto custo em uma geração, o gestor já tem uma situação de otimização, e se ele quiser diminuir o custo do combustível, haverá um grande aumento nas emissões. O ponto azul que aparece na Figura 5.7 e na Figura 5.8 representa a melhor opção encontrada em ambas as abordagens.

Para melhorar a análise no estudo de caso da usina de geração de energia com 10 geradores, foi feita uma comparação entre AC, AC+SA, AC+BT com AG clássico, tanto em relação ao custo de geração de energia quanto às emissões. As tabelas 5.22 e 5.23, mostram as respectivas comparações.

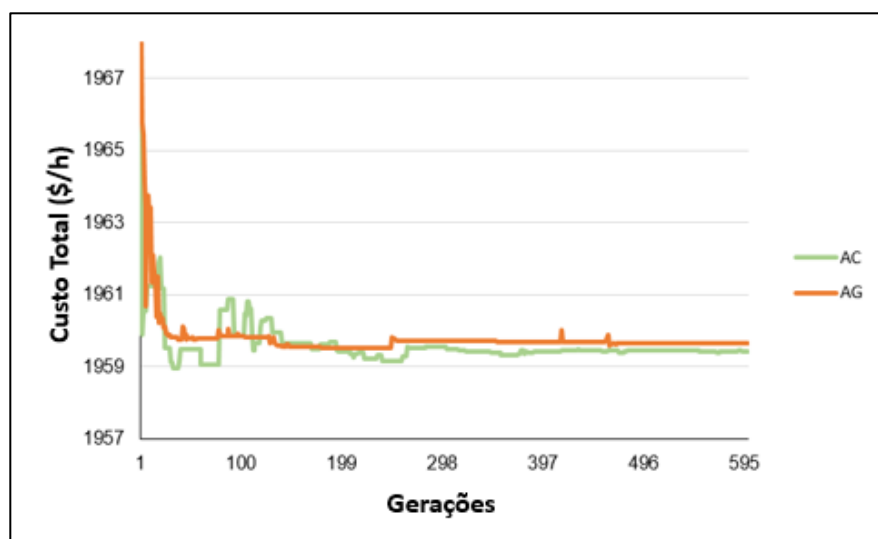
**Tabela 5.22:** Comparação de AC, AC+SA, AC+BT com o AG clássico (Custos).

Custo de Geração por Gerador em USD/h				
Soluções ELD:	AC Clássico	AC+SA	AC+BT	AG Clássico
Cost1	247.64	247.46	247.64	248.71
Cost2	237.13	237.13	236.93	235.93
Cost3	237.04	237.14	237.12	237.91
Cost4	221.82	222.17	221.82	220.62
Cost5	239.56	239.68	239.56	239.25
Cost6	142.11	142.39	141.99	139.35
Cost7	155.50	154.99	155.36	158.24
Cost8	156.07	156.01	155.65	159.02
Cost9	184.71	184.71	184.71	182.75
Cost10	137.84	137.84	137.84	137.86
Custo Total	1959.42	1959.52	<b>1958.62</b>	1959.64

**Tabela 5.23:** Comparação de AC, AC+SA, AC+BT com o AG clássico (Emissões).

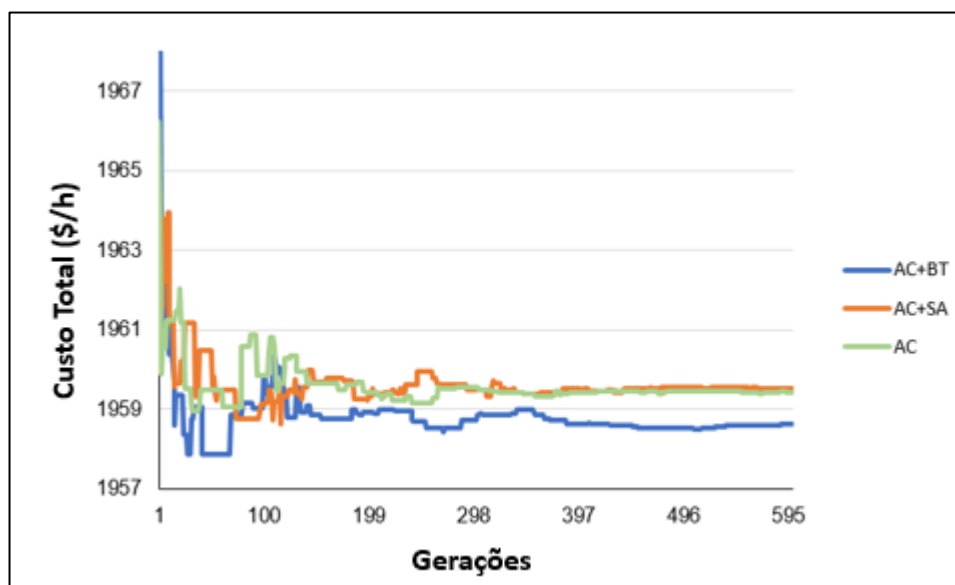
Emissões por Gerador em m <sup>3</sup> /h				
Soluções ELD:	AC Clássico	AC+SA	AC+BT	AG Clássico
Em1	89.20	85.20	89.20	115.70
Em2	190.95	190.94	188.90	178.86
Em3	159.98	161.91	161.59	176.62
Em4	156.79	161.95	156.79	140.06
Em5	147.46	149.28	147.46	142.83
Em6	144.29	148.06	142.72	110.45
Em7	141.16	135.53	139.61	173.16
Em8	147.46	146.79	142.72	182.91
Em9	163.73	163.73	163.73	152.15
Em10	13.17	13.17	13.17	13.25
Total de Emissões	1354.19	1356.56	<b>1345.89</b>	1386.99

Quanto aos custos e emissões, os valores encontrados no AG são piores que os encontrados no AC e suas hibridações (Tabela 5.23). Pode ser visto no gráfico da Figura 5.9 que, com o passar das gerações, a evolução no AG às vezes pára em alguns pontos e somente muda de situação quando ocorre uma mutação. No entanto, o AC realiza uma exploração mais diversificada, apoiando sua evolução na melhoria de suas fontes de conhecimento.

**Figura 5.9:** Convergência AG vs AC.

A Figura 5.10 mostra o comportamento de AC, AC+SA e AC+BT. Pode-se observar que a busca local realiza a exploração próxima a um ponto da vizinhança de um valor que para o AC é considerado ótimo.

O valor final encontrado em AC+BT é melhor que em AC e AC+SA, no entanto, nota-se que nas primeiras gerações os AC+BT, foram encontrados melhores valores do que seu último resultado.

**Figura 5.10:** Convergência AC, AC+BT e AC+SA.

A comparação do mínimo, média, custo máximo e desvio padrão dos algoritmos AC, AC+SA, AC+BT e AG para a aplicação desses algoritmos na planta de 10 geradores é mostrada na Tabela 5.24.

**Tabela 5.24:** Comparação estatística para os 10 geradores.

Algoritmo	Mínimo (\$/h)	Média (\$/h)	Máximo (\$/h)	Desvio Padrão
AC Clássico	1,959.42	1,959.46	1,959.49	0.034
AC+SA	1,959.52	1,959.55	1,959.58	0.031
AC+BT	1,958.62	1,958.66	1,958.68	0.030
AG	1,959.64	1,959.70	1,959.74	0.051

O uso de hibridização de SA com BT proporcionou um resultado muito próximo do AC+SA em relação ao custo. No entanto, o híbrido AC+BT obteve um melhor resultado. A Tabela 5.25, mostra o resultado dessa comparação em relação ao custo.

**Tabela 5.25:** Comparação de AC+SA, AC+BT com SA+BT (Custos)

Custo de Geração por Gerador em USD/h			
Soluções ELD:	AC+SA	AC+BT	SA+BT
Cost1	247.46	247.64	249.27
Cost2	237.13	236.93	234.23
Cost3	237.14	237.12	236.38
Cost4	222.17	221.82	219.98
Cost5	239.68	239.56	238.93
Cost6	142.39	141.99	146.41
Cost7	154.99	155.36	162.45
Cost8	156.01	155.65	154.76
Cost9	184.71	184.71	179.28
Cost10	137.84	137.84	137.85
Custo Total	1959.52	<b>1958.62</b>	1959.54

Quando os valores de emissão são observados, pode ser visto na tabela 5.26, que o resultado usando AC+SA e AC+BT são melhores do que aqueles obtidos com a hibridação SA+BT.

**Tabela 5.26:** Comparação de AC+SA, AC+BT com SA+BT (Emissões).

Emissões por Gerador em m <sup>3</sup> /h			
Soluções ELD:	AC+SA	AC+BT	SA+BT
Em1	85.20	89.20	130.94
Em2	190.94	188.90	162.42
Em3	161.91	161.59	147.94
Em4	161.95	156.79	131.44
Em5	149.28	147.46	138.06
Em6	148.06	142.72	206.15
Em7	135.53	139.61	228.77
Em8	146.79	142.72	133.06
Em9	163.73	163.73	132.69
Em10	13.17	13.17	13.20
Total de Emissões	1356.56	<b>1345.89</b>	1424.66

## B. RESULTADOS DO SISTEMA DE TESTE IEEE ELD COM 13 UNIDADES

A Tabela 5.27, mostra o resultado obtido pelo uso dos algoritmos propostos neste artigo com o desligamento dos motores mais caros na geração de energia elétrica. Não há comparação com outros resultados, devido ao fato de que não foram encontrados em outros trabalhos realizados com os mesmos parâmetros e abordagem deste trabalho (SINHA *et al.*, 2003).

**Tabela 5.27:** Resultados deste trabalho aplicado sistema de teste IEEE com 13 geradores.

Algoritmo	AC Desligando geradores	AC+SA Desligando geradores	AC+BT Desligando geradores
Custo Total USD/h	23683.83	23683.83	<b>23683.82</b>
Total de Emissões m <sup>3</sup> /h	29593.17	<b>29592.83</b>	29592.87

O menor custo total e a menor emissão ocorrem nessa abordagem, quando AC+BT e AC+SA são usados, respectivamente. Estes valores são mostrados em azul na Tabela 5.27.

Os resultados obtidos a partir da aplicação dos algoritmos propostos neste artigo, resolvendo o problema do CEELD no sistema IEEE com 13 geradores, foram comparados com alguns outros resultados encontrados na literatura científica. A Tabela 5.28 apresenta essa comparação.

**Tabela 5.28:** Algoritmo proposto vs outros.

Algoritmo	Custo Total USD/h	Total de Emissões m <sup>3</sup> /h
<b>AC Clássico</b>	<b>24133.50</b>	<b>27811.43</b>
<b>AC+SA</b>	<b>24098.78</b>	<b>27749.19</b>
<b>AC+BT</b>	<b>24046.34</b>	<b>27655.95</b>
<b>SA+BT</b>	<b>24393.02</b>	<b>16397.39</b>
BGA(DAS e PATVARDHAN, 1998)	24703.32	29480.00
IGA(DAS e PATVARDHAN, 1998)	24398.63	29500.00
GAA(DAS e PATVARDHAN, 1998)	24418.99	29500.00
GAA2(DAS e PATVARDHAN, 1998)	24398.23	29500.00
SABED(DAS e PATVARDHAN, 1998)	24970.91	29580.00
MOSST(DAS e PATVARDHAN, 1998)	24261.00	28970.00
DE(NOMAN e IBA, 2008)	24169.92	-
IGWO(MEHMOOD e AHMAD, 2018)	24202.16	-

NRHS(AL-BETAR <i>et al.</i> , 2018)	24164.06	-
MTS (ARUL <i>et al.</i> , 2014)	24169.63	-
HS(ARUL <i>et al.</i> , 2014)	24208.70	-
HHS(ARUL <i>et al.</i> , 2014)	24169.90	-
IHS(ARUL <i>et al.</i> , 2014)	24164.32	-

A comparação dos dados estatísticos do valor mínimo, médio, custo máximo e desvio padrão dos algoritmos AC, AC+SA e AC+BT propostos neste trabalho com os resultados de outros métodos recentemente reportados é apresentada na Tabela 5.29.

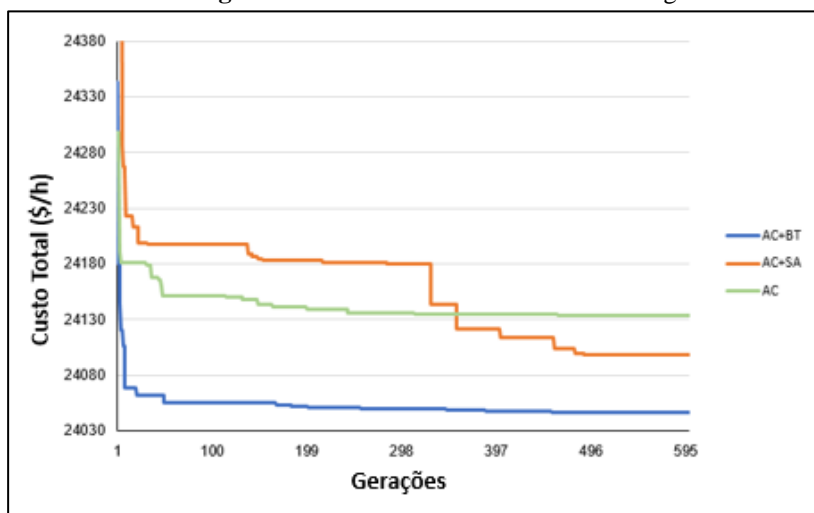
**Tabela 5.29:** Comparação estatística dos dados.

Algoritmo	Mínimo (\$/h)	Média (\$/h)	Máximo (\$/h)	Desvio Padrão
IHS(ARUL <i>et al.</i> , 2014)	24,164.32	24,164.94	24,166.54	1.854
HS (ARUL <i>et al.</i> , 2014)	24,208.70	24,323.20	24,503.70	-
HHS (ARUL <i>et al.</i> , 2014)	24,169.90	24,169.90	24,169.90	-
MTS (ARUL <i>et al.</i> , 2014)	24,169.63	24,179.26	24,196.74	7.59
IGWO(MEHMOOD e AHMAD, 2018)	24,202.16	24,210.00	24,228.35	7.021
NRHS(AL-BETAR <i>et al.</i> , 2018)	24,164.06	24,185.61	-	-
<b>SA+BT</b>	<b>24,393.02</b>	<b>24,395.02</b>	<b>24,398.03</b>	<b>2.649</b>
<b>AC Clássico</b>	<b>24,133.50</b>	<b>24,133.69</b>	<b>24,133.83</b>	<b>0.170</b>
<b>AC+SA</b>	<b>24,098.48</b>	<b>24,098.78</b>	<b>24,098.91</b>	<b>0.222</b>
<b>AC+BT</b>	<b>24,046.34</b>	<b>24,046.44</b>	<b>24,046.62</b>	<b>0.160</b>

Os desvios-padrão encontrados nos algoritmos deste artigo foram menores que os apresentados na Tabela 5.29, e o algoritmo AC+BT apresentou o menor valor em relação aos demais.

A curva característica de convergência de cada algoritmo proposto neste artigo é mostrada na Figura 5.11, que corresponde à solução ótima relatada na Tabela 5.29.

**Figura 5.11:** Curva característica de convergência.



A característica de convergência do algoritmo AC+BT proposto é rápida e suave para alcançar a solução ótima conforme projetada na Tabela 5.28.

### 5.3.2 Engenharia Mecânica

Os resultados obtidos em todos os problemas abordados foram comparados com o trabalho de outros autores e mostraram-se melhores.

#### A. Projeto de Mola Helicoidal de Compressão

Os resultados da simulação AC+BT para 10 iterações, população de 50 indivíduos, 300 gerações e tamanho da lista tabu de 6 são mostrados na Tabela 5.30.

**Tabela 5.30:** Melhores soluções para a mola de compressão helicoidal.

Variáveis de Projeto	Yan et al (2012)	Belegundu (1982)	Arora (2004)	Coello (2000)	Coello and Montes (2002)	Este Trabalho
$x_1(d)$	0.051728	0.050000	0.0519	0.051480	0.051989	0.052292
$x_2(D)$	0.357644	0.315900	0.3620	0.351661	0.363965	0.371421
$x_3(P)$	11.244543	14.250000	11.0000	11.632201	10.890522	10.476284
$g_1(x)$	-0.000845	-0.000014	-0.001879	-0.002080	-0.000013	-0.000073
$g_2(x)$	-1.2600e-05	-0.003782	-0.132585	-0.000110	-0.000021	-0.130655
$g_3(x)$	-4.051300	-3.938302	-4.056841	-4.026318	-4.061338	-4.081791
$g_4(x)$	-0.727090	-0.756067	-0.724067	-4.026318	-0.722698	-0.717525
$f(x)$	0.0126747	0.0128334	0.0126761	0.0127048	0.0126810	0.0126713

**Fonte:** (BELEGUNDU e ARORA, 1982; COELLO, 2000; COELLO e MONTES, 2002; ARORA, 2004; YAN *et al.*, 2012).

O trabalho de Yan et al (2012) utilizou o AC, no entanto, quando o BT foi incluído no AC, obtivemos um resultado ainda melhor, uma vez que o BT buscou um melhor valor na vizinhança do valor encontrado pela AC.

#### B. Projeto do vaso de pressão

Os resultados da simulação AC+BT para 10 iterações, população de 50 indivíduos, 300 gerações e tamanho da lista tabu de 6 são mostrados na Tabela 5.31.

**Tabela 5.31:** Comparação do AC+BT com outros algoritmos (melhor resultado encontrado para o Problema do Recipiente de Pressão)

Variáveis de Projeto	AC-BL (SILVA, 2012)	AC (SILVA, 2012)	GQPSO (1) (COELHO, 2010)	COELLO, MONTES, 2002)	CPSO (HE, WANG, 2007)	SC-ABC (BRAJEVIC; TUBA; SUBOTIC, 2011)	Este Trabalho
X <sub>1</sub> (TS)	0.81250	0.8125000	0.8125	0.812500	0.812500	0.81250	0.8125
X <sub>2</sub> (TH)	0.43750	0.4375000	0.4375	0.437500	0.437500	0.43750	0.4375
X <sub>3</sub> (R)	42.098423	42.098414	42.0984	42.097398	42.091266	42.098187	42.098445
X <sub>4</sub> (L)	176.636897	176.636986	176.6372	176.654050	176.746500	176.640750	176.636669
g <sub>1</sub> (x)	-4.361E-07	-6,098E-07	-8.7999E-7	-0.000020	-0.000139	-4.988451	-1,15E-08
g <sub>2</sub> (x)	-0.0358810	-0,03588113	-3.5881E-2	-0.035891	-0.035949	-0.035883	-0,035881
g <sub>3</sub> (x)	-0,11781066	0,007605214	-0.2179	-27.886075	-116.382700	-5.297613	-0,366215
g <sub>4</sub> (x)	-63.36310	-63,36301	-63.3628	-63.345953	-63.253500	-63.359250	-63,363331
f(x)	6059.7177	6059.7181	6059.7208	6059.9463	6061.0777	6059.768058	6059,7159

Fonte: Adaptado de (SILVA, 2012a).

Os resultados mais recentes da Tabela 5.31 estão utilizando o AC e o AC-BL (foi utilizado o recozimento simulado em busca local), apresentado no trabalho de Silva (2012). O resultado do nosso estudo com o AC+BT atingiu um valor melhor para a função objetiva f(x).

### C. Redutor de Velocidade

Os resultados da simulação AC+BT para 10 iterações, população de 50 indivíduos, 300 gerações e tamanho da lista tabu de 6 são mostrados na Tabela 5.32.

**Tabela 5.32:** Resultados da otimização do redutor de velocidade.

Variáveis de Projeto	Brajevic et al (2010)	Cagnina et al, (2008)	Este Trabalho
x1	3,500000	3,500000	3,500962
x2	0,700000	0,700000	0,700000
x3	17,000000	17,000000	17,000000
x4	7,300000	7,300000	7,300000
x5	7,800000	7,800000	7,800000
x6	3,350215	3,350214	3,350358
x7	5,286683	5,286683	5,287909
g <sub>1</sub> (x)	-0,073915	-0,073915	-0,074170
g <sub>2</sub> (x)	-0,197996	-0,197998	-0,675535
g <sub>3</sub> (x)	-0,499172	-0,499172	-0,499258
g <sub>4</sub> (x)	-0,901471	-0,901471	-0,901563
g <sub>5</sub> (x)	-2,220E-16	0,000000	-0,000046
g <sub>6</sub> (x)	-3,331E-16	-5,000E-16	-0,000685
g <sub>7</sub> (x)	-0,702500	-0,702500	-0,702500



g8 (x)	0,000000	-1,000E-16	-0,000275
g9 (x)	-0,583333	-0,583333	-0,583219
g10 (x)	-0,051326	-0,051325	-0,051296
g11 (x)	-0,010852	-0,010852	-0,010680
f (x)	2996,348165	2996,348165	2996,084011

Fonte: (CAGNINA *et al.*, 2008; BRAJEVIC *et al.*, 2010).

Para este problema os resultados encontrados em nossa proposta de uso do AC+BT foram melhores que os resultados da tabela 5.32.

### 5.3.3 Engenharia Civil

#### A. Projeto de coluna tubular

Os resultados da simulação AC+BT para 10 iterações, população de 50 indivíduos, 300 gerações e tamanho da lista tabu igual a 6, são mostrados na Tabela 5.33.

**Tabela 5.33:** Melhores soluções para o exemplo da coluna tubular.

Variáveis de Projeto	Rao (1996)	Hsu, Liu (2007)	Gandomi et al. (2013b)	Rocha, Fernandes (2009)	Gandomi et al. (2015)	Este Trabalho
d	5.44	5.4507	5.45139	5.451083	5.451278	5.453984
T	0.293	0.292	0.29196	0.29199	0.291957	0.291814
g <sub>1</sub>	-0.8579	-0.00008	-0.0241	-0.00007	0.00000	0.00000
g <sub>2</sub>	<b>0.0026<sup>a</sup></b>	<b>0.1317<sup>a</sup></b>	-0.1095	-0.00004	-0.00004	-0.00103
f(d,t)	26.5323	25.5316	26.53217	26.53227	26.5314	26.50515

<sup>a</sup>Negrito são os valores violados.

Fonte: (HSU e LIU, 2007; RAO e RAO, 2009; ROCHA e FERNANDES, 2009; GANDOMI *et al.*, 2013; GANDOMI e ALAVI, 2015)

O resultado do AC+BT foi maior que o obtido pelas obras que estão na tabela 5.33. Percebe-se que os trabalhos das duas primeiras colunas da tabela 5.33, violaram a restrição g<sub>2</sub>. Em nossa solução obtivemos o valor 0 para a restrição g<sub>1</sub>, mas com este valor a restrição não é violada.

## 6 CONCLUSÃO E RECOMENDAÇÕES PARA TRABALHOS FUTUROS

### 6.1 Conclusão

No desenvolvimento deste trabalho, foram utilizadas as técnicas heurísticas: busca tabu (BT), recozimento simulado (SA), escalando a colina (HC) e a busca em feixes (BS), como busca local para melhorar os resultados obtidos pelo algoritmo cultural (AC), apesar do AC já proporcionar bons resultados quando aplicado na solução de problemas de otimização. Desta forma, foram elaborados quatro algoritmos meméticos pela hibridização do AC com cada uma das técnicas de busca local utilizadas. Na hibridização do AC com técnicas BT, SA, BS e HC, foram utilizados os algoritmos básicos destas técnicas, com variação de parâmetros das três primeiras técnicas. Estas variações proporcionaram vários cenários para avaliação dos resultados entre os algoritmos criados por cada hibridização.

O AC chamado de clássico neste trabalho possui uma inovação quanto a forma de alimentar o conhecimento geográfico. Pois, este seleciona os três melhores resultados para definir uma área de prováveis soluções ótimas. Este mesmo AC foi utilizado na hibridização.

Os resultados obtidos nas aplicações dos algoritmos meméticos desenvolvidos em funções multimodais, serviram para destacar os que tiveram melhor desempenho em certos conjuntos de funções. Quando utilizado para resolver funções básicas o teste não paramétrico Hellinger-TOPSIS, mostrou que os meméticos foram melhores que o AC clássico quando o domínio possuía 10 e 30 variáveis.

Quando a abordagem incluiu as funções híbridas os testes de Friedman, Friedman Aligned e Quade, mostram que os algoritmos meméticos obtiveram valores melhores que o AC clássico. As funções híbridas, deram a base para a solução dos problemas reais com restrições.

Na aplicação em engenharia, os algoritmos desenvolvidos nessa tese se mostraram superiores em relação aos resultados obtidos em trabalhos da mesma natureza.

O problema de engenharia elétrica de despacho de energia e emissões de gases em uma planta real, já tinha sido resolvida por outros dois autores que utilizaram o DE e o SA, e quando resolvido pelo AC clássico com a nova técnica de alimentação do conhecimento topográfico, este último obteve melhores resultados. Porém, o uso dos meméticos: AC+SA e AC+BT, foram ainda melhores que o AC. Este mesmo problema quando resolvido com o AG, teve solução pior que quando resolvido pelos algoritmos: AC, AC+SA e AC+BT.

Ainda abordando o problema de despacho de energia e emissões de gases com o uso do sistema de teste do IEEE com 13 geradores, foi criado neste trabalho um memético da hibridização do SA+BT, que obteve o melhor resultado em relação as emissões de gases, e o AC+BT teve o menor custo total de combustíveis. Os meméticos deste trabalho foram comparados com outras publicações internacionais na solução do mesmo problema no mesmo cenário e os resultados foram sempre melhores.

Na aplicação de engenharia mecânica e civil, abordada nesta tese, foram utilizados três problemas clássicos já resolvidos por outros autores, e o comportamento dos algoritmos aqui apresentados obtiveram valores iguais ou superiores aos pesquisados neste trabalho.

Com isso, nota-se que a hibridização do algoritmo cultural com técnica de busca local, fornece um algoritmo memético, que resulta em melhores soluções quando aplicados a problemas com variáveis reais.

Esses resultados mostram que a evolução dos conhecimentos dos indivíduos impacta de forma positiva na melhoria do desempenho da busca e que a busca local melhora ainda mais esse resultado em determinadas situações.

## 6.2 *Recomendações para trabalhos futuros*

Como sugestões para trabalhos futuros têm-se:

- Investigar o uso de mais de uma técnica de busca local na intensificação da pesquisa na vizinhança do ponto encontrado na busca global;
- Incrementar esta pesquisa com a aplicação dos demais conhecimentos do AC;
- Realizar a comparação desta pesquisa com substituição dos conhecimentos do AC, para verificar os impactos na utilização de cada conhecimento;
- Aplicar nesta pesquisa a variação de outros parâmetros das buscas locais.

### 6.3 Publicações geradas pela tese

Esta tese durante seu desenvolvimento originou 3 (três) artigos:

**1- Algoritmo cultural com busca local avaliado através de testes estatísticos não paramétricos.**

Aceito para apresentação no **XIII Brazilian Congress on Computational Intelligence – CBIC 2017**, Niterói, RJ, Brasil, 2017.

Disponível em: <http://cbic2017.org/papers/cbic-paper-76.pdf>

**2- Cultural algorithm with local search evaluated through non-parametric statistical tests.**

Publicado na revista **Journal of Engineering and Technology for Industrial Applications**, ISSN 2447-0228.

Disponível em: <https://dx.doi.org/10.5935/2447-0228.20170072>

**3- Solution to Economic – Emission Load Dispatch by Cultural Algorithm Combined with Local Search: Case Study.**

Publicado na revista **IEEE ACCESS**, ISSN 2169-3536, Classificação Qualis A2 em Engenharia IV.

Disponível em: <https://dx.doi.org/10.1109/ACCESS.2018.2877770>

## 7 REFERÊNCIAS

AL-BETAR, M. A. et al. **Economic load dispatch problems with valve-point loading using natural updated harmony search.** Neural Computing and Applications, v. 29, n. 10, p. 767-781, 2018. ISSN 0941-0643.

ALI, M. Z.; AWAD, N.; REYNOLDS, R. G. **Balancing search direction in cultural algorithm for enhanced global numerical optimization.** Swarm Intelligence (SIS), 2014 IEEE Symposium on, 2014, IEEE. p.1-7.

ALI, M. Z.; AWAD, N. H. **A novel class of niche hybrid cultural algorithms for continuous engineering optimization.** information sciences, v. 267, p. 158-190, 2014. ISSN 0020-0255.

ALTUN, H.; YALCINOZ, T. **Implementing soft computing techniques to solve economic dispatch problem in power systems.** Expert Systems with Applications, v. 35, n. 4, p. 1668-1678, 2008. ISSN 0957-4174.

ARMENTANO, V. A.; BRANCHINI, R. M. **Uma Introdução à Busca Tabu.** 2013.

ARORA, J. **Introduction to optimum design.** Academic Press, 2004. ISBN 0080470254.

ARUL, R.; RAVI, G.; VELUSAMI, S. **An improved harmony search algorithm to solve economic load dispatch problems with generator constraints.** Electrical Engineering, v. 96, n. 1, p. 55-63, 2014. ISSN 0948-7921.

AWAD, N. et al. **Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization.** 2016.

BACCI, T.; MATTIA, S.; VENTURA, P. **The Bounded Beam Search algorithm for the Block Relocation Problem.** Computers & Operations Research, 2018. ISSN 0305-0548.

BARNDORFF-NIELSEN, O. E. **Parametric statistical models and likelihood.** Springer Science & Business Media, 2012. ISBN 1461239346.

BECCENERI, J. C. **Meta-heurísticas e Otimização Combinatória: Aplicações em Problemas Ambientais**. INPE, Sao José dos Campos, 2008.

BEHESHTI, Z.; SHAMSUDDIN, S. M. H. **A review of population-based meta-heuristic algorithms**. Int. J. Adv. Soft Comput. Appl, v. 5, n. 1, p. 1-35, 2013.

BEHZADIAN, M. et al. **A state-of the-art survey of TOPSIS applications**. Expert Systems with applications, v. 39, n. 17, p. 13051-13069, 2012. ISSN 0957-4174.

BELEGUNDU, A. D.; ARORA, J. S. **A study of mathematical programming methods for structural optimization. Part I: Theory**. International Journal for Numerical Methods in Engineering, v. 21, n. 9, p. 1583-1599, 1982. ISSN 1097-0207.

BENAVOLI, A.; CORANI, G.; MANGILI, F. **Should we really use post-hoc tests based on mean-ranks?** The Journal of Machine Learning Research, v. 17, n. 1, p. 152-161, 2016. ISSN 1532-4435.

BENNEL, J. A.; CABO, M.; MARTINEZ-SYKORA, A. **A beam search approach to solve the convex irregular bin packing problem with guillotine cuts**. European Journal of Operational Research, v. 270, n. 1, p. 89-102, 2018. ISSN 0377-2217.

BHUVANA, J.; ARAVINDAN, C. **Memetic algorithm with preferential local search using adaptive weights for multi-objective optimization problems**. Soft Computing, v. 20, n. 4, p. 1365-1388, 2016. ISSN 1432-7643.

BLUM, C. et al. **Hybrid metaheuristics in combinatorial optimization: A survey**. Applied Soft Computing, v. 11, n. 6, p. 4135-4151, 2011. ISSN 1568-4946.

BRAJEVIC, I.; TUBA, M.; SUBOTIC, M. **Improved artificial bee colony algorithm for constrained problems**. Proceedings of the 11th WSEAS international conference on neural networks and 11th WSEAS international conference on evolutionary computing and 11th WSEAS international conference on Fuzzy systems, 2010, World Scientific and Engineering Academy and Society (WSEAS). p.185-190.

BROWNLEE, J. **Clever algorithms: nature-inspired programming recipes**. Jason Brownlee, 2011. ISBN 1446785068.

BURKE, E.; SMITH, A. **A memetic algorithm to schedule planned grid maintenance**. Proc. Int. Conf. on Computational Intelligence for Modelling Control and Automation (Vienna, Austria), 1999. p.122-127.

CAGNINA, L. C.; ESQUIVEL, S. C.; COELLO, C. A. C. **Solving engineering optimization problems with the simple constrained particle swarm optimizer**. Informatica, v. 32, n. 3, 2008. ISSN 1854-3871.

CAMPOS, G. M. **Estatística prática para docentes e pós-graduandos**. São Paulo: Faculdade de Odontologia de Ribeirão Preto, 2001.

CHAMBERS, L. D. **Practical handbook of genetic algorithms: complex coding systems**. CRC press, v. Vol III, 1999.

CHEN, Z.; LI, S.; YUE, W. **Memetic algorithm-based multi-objective coverage optimization for wireless sensor networks**. *Sensors*, v. 14, n. 11, p. 20500-20518, 2014.

COELHO, L. D. S.; ALOTTO, P. **Electromagnetic optimization using a cultural self-organizing migrating algorithm approach based on normative knowledge**. *IEEE Transactions on Magnetics*, v. 45, n. 3, p. 1446-1449, 2009. ISSN 0018-9464.

COELLO, C. A. C. **Use of a self-adaptive penalty approach for engineering optimization problems**. *Computers in Industry*, v. 41, n. 2, p. 113-127, 2000. ISSN 0166-3615.

COELLO, C. A. C.; MONTES, E. M. **Constraint-handling in genetic algorithms through the use of dominance-based tournament selection**. *Advanced Engineering Informatics*, v. 16, n. 3, p. 193-203, 2002. ISSN 1474-0346.

DAS, D. B.; PATVARDHAN, C. **New multi-objective stochastic search technique for economic load dispatch**. *IEE Proceedings-Generation, Transmission and Distribution*, v. 145, n. 6, p. 747-752, 1998. ISSN 1359-7051.

DAWKINS, R. **O gene egoísta**. Editora Companhia das Letras, 2017. ISBN 8543810094.

DE AZEVEDO, A. T.; RIBEIRO, C. M.; TEIXEIRA JR, R. F. **Um algoritmo do tipo Beam Search para alocação de células a centrais de telefonia celular**. *Revista GEPROS*, v. 8, n. 2, p. 9, 2013. ISSN 1984-2430.

DE JONG, K. A. **Analysis of the behavior of a class of genetic adaptive systems**. 1975.

DE JONG, K. A. **Evolutionary computation: a unified approach**. MIT press, 2006. ISBN 0262041944.

DECERLE, J. et al. **A memetic algorithm for multi-objective optimization of the home health care problem**. *Swarm and evolutionary computation*, v. 44, p. 712-727, 2019. ISSN 2210-6502.

DERRAC, J. et al. **Analyzing convergence performance of evolutionary algorithms: A statistical approach**. *Information Sciences*, v. 289, p. 41-58, 2014. ISSN 0020-0255.

DERRAC, J. et al. **A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms**. *Swarm and Evolutionary Computation*, v. 1, n. 1, p. 3-18, 2011. ISSN 2210-6502.

DIGALAKIS, J. G.; MARGARITIS, K. G. **A multipopulation cultural algorithm for the electrical generator scheduling problem**. *Mathematics and Computers in Simulation*, v. 60, n. 3-5, p. 293-301, 2002. ISSN 0378-4754.

FENG, L.; ONG, Y.-S.; GUPTA, A. **Genetic Algorithm and Its Advances in Embracing Memetics**. In: (Ed.). *Evolutionary and Swarm Intelligence Algorithms*: Springer, 2019. p.61-84.

FIGUEREDO, G. P.; BERNARDINO, H. S.; BARBOSA, H. J. C. **Introdução aos sistemas imunológicos artificiais**. Meta-heurísticas em pesquisa operacional, p. 113-128, 2013.

GABRIEL, P.; DELBEM, A. **Fundamentos de algoritmos evolutivos, Relatório técnico**. Notas Didáticas do ICMC-USP, v. 75, 2008.

GANDOMI, A. H.; ALAVI, A. H. **An introduction of krill herd algorithm for engineering optimization**. Journal of Civil Engineering and Management, v. 22, n. 3, p. 302-310, 2015. ISSN 1392-3730.

GANDOMI, A. H.; YANG, X.-S.; ALAVI, A. H. **Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems**. Engineering with computers, v. 29, n. 1, p. 17-35, 2013. ISSN 0177-0667.

GARCÍA, S. et al. **A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization**. Journal of Heuristics, v. 15, n. 6, p. 617-644, 2009. ISSN 1381-1231.

GASPAR-CUNHA, A.; TAKAHASHI, R.; ANTUNES, C. H. **Manual de computação evolutiva e metaheurística**. Imprensa da Universidade de Coimbra/Coimbra University Press, 2012. ISBN 9892601505.

GIBBONS, J. D.; CHAKRABORTI, S. **Nonparametric statistical inference**. In: (Ed.). International encyclopedia of statistical science: Springer, 2011. p.977-979.

GOGNA, A.; TAYAL, A. **Metaheuristics: review and application**. Journal of Experimental & Theoretical Artificial Intelligence, v. 25, n. 4, p. 503-526, 2013. ISSN 0952-813X.

GOLBERG, D. E. **Genetic algorithms in search, optimization, and machine learning**. Addison Wesley, v. 1989, n. 102, p. 36, 1989.

GONG, X. et al. **Energy and Labor Aware Production Scheduling for Industrial Demand Response Using Adaptive Multi-objective Memetic Algorithm**. IEEE Transactions on Industrial Informatics, 2018. ISSN 1551-3203.

GOUDARZI, A. et al. **Non-convex optimization of combined environmental economic dispatch through the third version of the cultural algorithm (CA3)**. Power and Energy Conference (TPEC), IEEE Texas, 2017a, IEEE. p.1-6.

\_\_\_\_\_. **Non-convex optimization of combined environmental economic dispatch through the third version of the cultural algorithm (CA3)**. 2017 IEEE Texas Power and Energy Conference (TPEC), 2017b, IEEE. p.1-6.

GRÉWAL, G.; COROS, S.; VENTRESCA, M. **A Memetic Algorithm for Performing Memory Assignment in Dual-Bank DSPs**. International Journal of Computational Intelligence and Applications, v. 6, n. 04, p. 473-497, 2006. ISSN 1469-0268.



GUAN, J.; LIN, G.; FENG, H.-B. **A learning-based probabilistic tabu search for the uncapacitated single allocation hub location problem.** Computers & Operations Research, v. 98, p. 1-12, 2018. ISSN 0305-0548.

HART, W. E. **Adaptive global optimization with local search.** 1994. Citeseer

HEMMATI, R.; AZIZI, N. **Optimal control strategy on battery storage systems for decoupled active-reactive power control and damping oscillations.** Journal of Energy Storage, v. 13, p. 24-34, 2017. ISSN 2352-152X.

HSU, Y.-L.; LIU, T.-C. **Developing a fuzzy proportional–derivative controller optimization engine for engineering design optimization problems.** Engineering Optimization, v. 39, n. 6, p. 679-700, 2007. ISSN 0305-215X.

JIA, S.; HU, Z.-H. **Path-relinking Tabu search for the multi-objective flexible job shop scheduling problem.** Computers & Operations Research, v. 47, p. 11-26, 2014. ISSN 0305-0548.

JIN, X.; REYNOLDS, R. G. **Using knowledge-based systems with hierarchical architectures to guide evolutionary search.** International Journal on Artificial Intelligence Tools, v. 9, n. 01, p. 27-44, 2000. ISSN 0218-2130.

JÚNIOR, J. D. A. B. et al. **Solution to economic emission load dispatch by simulated annealing: case study.** Electrical Engineering, p. 1-13, 2017. ISSN 0948-7921.

KATSIGIANNIS, Y. A.; GEORGILAKIS, P. S.; KARAPIDAKIS, E. S. **Hybrid simulated annealing–tabu search method for optimal sizing of autonomous power systems with renewables.** IEEE Transactions on Sustainable Energy, v. 3, n. 3, p. 330-338, 2012. ISSN 1949-3029.

KATSIGIANNIS, Y. A.; STAVRAKAKIS, G. S. **Estimation of wind energy production in various sites in Australia for different wind turbine classes: A comparative technical and economic assessment.** Renewable energy, v. 67, p. 230-236, 2014. ISSN 0960-1481.

KIM, S.-S.; SMITH, A. E.; LEE, J.-H. **A memetic algorithm for channel assignment in wireless FDMA systems.** Computers & operations research, v. 34, n. 6, p. 1842-1856, 2007. ISSN 0305-0548.

KOBTI, Z. **Heterogeneous multi-population cultural algorithm.** Evolutionary Computation (CEC), 2013 IEEE Congress on, 2013, IEEE. p.292-299.

KÓCZY, L. T.; FÖLDESI, P.; TŰŰ-SZABÓ, B. **Enhanced discrete bacterial memetic evolutionary algorithm-an efficacious metaheuristic for the traveling salesman optimization.** Information Sciences, v. 460, p. 389-400, 2018. ISSN 0020-0255.

KRASNOGOR, N.; SMITH, J. **A tutorial for competent memetic algorithms: model, taxonomy, and design issues.** IEEE Transactions on Evolutionary Computation, v. 9, n. 5, p. 474-488, 2005. ISSN 1089-778X.

KROHLING, R. A.; LOURENZUTTI, R.; CAMPOS, M. **Ranking and comparing evolutionary algorithms with Hellinger-TOPSIS**. Applied Soft Computing, v. 37, p. 217-226, 2015. ISSN 1568-4946.

KU, K. W.; MAK, M.-W. **Empirical analysis of the factors that affect the Baldwin effect**. International Conference on Parallel Problem Solving from Nature, 1998, Springer. p.481-490.

KYRIACOU, S.; SARMA, P.; HUNT, I. **Constrained, Multi-Objective, Steamflood Injection Redistribution Optimization, using a Cloud-Distributed, Metamodel-Assisted, Memetic Optimization Algorithm**. SPE Reservoir Characterisation and Simulation Conference and Exhibition, 2017, Society of Petroleum Engineers.

LAI, X. et al. **A two-phase tabu-evolutionary algorithm for the 0–1 multidimensional knapsack problem**. Information Sciences, v. 436, p. 282-301, 2018. ISSN 0020-0255.

LIAO, G.-C.; TSAO, T.-P. **The use of genetic algorithm/fuzzy system and tabu search for short-term unit commitment**. Proceedings. International Conference on Power System Technology, 2002, IEEE. p.2302-2307.

LIN, G.; ZHU, W.; ALI, M. M. **An effective hybrid memetic algorithm for the minimum weight dominating set problem**. IEEE Transactions on Evolutionary Computation, v. 20, n. 6, p. 892-907, 2016. ISSN 1089-778X.

LIN, W.-M.; CHENG, F.-S.; TSAY, M.-T. **An improved tabu search for economic dispatch with multiple minima**. IEEE Transactions on power systems, v. 17, n. 1, p. 108-112, 2002. ISSN 0885-8950.

LIN, Y.; BIAN, Z.; LIU, X. **Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing–tabu search algorithm to solve the symmetrical traveling salesman problem**. Applied Soft Computing, v. 49, p. 937-952, 2016. ISSN 1568-4946.

LINDEN, R. **Algoritmos genéticos (2a edição)**. Brasport, 2008. ISBN 8574523739.

LIU, S.; CHEN, D.; WANG, Y. **Memetic algorithm for multi-mode resource-constrained project scheduling problems**. Journal of Systems Engineering and Electronics, v. 25, n. 4, p. 609-617, 2014. ISSN 1004-4132.

LUO, J. et al. **A hybrid multi-objective PSO–EDA algorithm for reservoir flood control operation**. Applied Soft Computing, v. 34, p. 526-538, 2015. ISSN 1568-4946.

MATSUMURA, T. et al. **A parallel tabu search and its hybridization with genetic algorithms**. Parallel Architectures, Algorithms and Networks, 2000. I-SPAN 2000. Proceedings. International Symposium on, 2000, IEEE. p.18-22.

MEHMOOD, K.; AHMAD, A. **Improved Grey Wolf Optimization for Economic Load Dispatch Problem Considering Valve Point Loading Effect and Prohibited Operating Zones**. The Nucleus, v. 54, n. 4, p. 250-257, 2018. ISSN 2306-6539.

MIGUEL, F. et al. **A memetic algorithm for the integral OBP/OPP problem in a logistics distribution center**. *Uncertain Supply Chain Management*, v. 7, n. 2, p. 203-214, 2019.

MIRSALEH, M. R.; MEYBODI, M. R. **Assignment of cells to switches in cellular mobile network: a learning automata-based memetic algorithm**. *Applied Intelligence*, p. 1-17, 2018. ISSN 0924-669X.

MOSCATO, P. **On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms**. *Caltech concurrent computation program, C3P Report*, v. 826, p. 1989, 1989.

MOSCATO, P.; COTTA, C.; MENDES, A. **Memetic algorithms**. In: (Ed.). *New optimization techniques in engineering*: Springer, 2004. p.53-85.

NAITALI, A.; GIRI, F. **Wiener and Hammerstein nonlinear systems identification using hybrid Genetic and Swarming Intelligence based Culture Algorithm**. *American Control Conference (ACC)*, 2010, IEEE. p.4528-4533.

NAJI, N. **A Review of the Metaheuristic Algorithms and their Capabilities (Particle Swarm Optimization, Firefly and Genetic Algorithms)**. 2017.

NALEPA, J.; BLOCHO, M. **Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows**. *Soft Computing*, v. 20, n. 6, p. 2309-2327, 2016. ISSN 1432-7643.

NARANG, N.; SHARMA, E.; DHILLON, J. **Combined heat and power economic dispatch using integrated civilized swarm optimization and Powell's pattern search method**. *Applied Soft Computing*, v. 52, p. 190-202, 2017. ISSN 1568-4946.

NASCIMENTO, M. H. R. et al. **A new solution to the economical load dispatch of power plants and optimization using differential evolution**. *Electrical Engineering*, p. 1-11, 2016. ISSN 0948-7921.

NERI, F.; KHAN, N. **Two local search components that move along the axes for memetic computing frameworks**. *Foundations of Computational Intelligence (FOCI)*, 2014 IEEE Symposium on, 2014, IEEE. p.62-69.

NGUYEN, T. T.; YAO, X. **Hybridizing cultural algorithms and local search**. *International Conference on Intelligent Data Engineering and Automated Learning*, 2006, Springer. p.586-594.

NOMAN, N.; IBA, H. **Differential evolution for economic load dispatch problems**. *Electric Power Systems Research*, v. 78, n. 8, p. 1322-1331, 2008. ISSN 0378-7796.

NORMAN, M. G.; MOSCATO, P. **A competitive and cooperative approach to complex combinatorial search**. *Proceedings of the 20th Informatics and Operations Research Meeting*, 1991, Citeseer. p.3.15-3.29.

NORVIG, P.; RUSSELL, S. **Inteligência Artificial: Tradução da 3a Edição**. Elsevier Brasil, 2014. ISBN 8535251413.

ODA, T. et al. **Analysis of mesh router placement in wireless mesh networks using friedman test**. 2014 IEEE 28th International Conference on Advanced Information Networking and Applications, 2014, IEEE. p.289-296.

OHATA, A. F. S. M. **Atribuição de turmas para professores via Beam Search em Java**. 2012.

OSTROWSKI, D. A.; REYNOLDS, R. G. **Knowledge-based software testing agent using evolutionary learning with cultural algorithms**. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), 1999, IEEE. p.1657-1663.

OUENNICHE, J.; PÉREZ-GLADISH, B.; BOUSLAH, K. **An out-of-sample framework for TOPSIS-based classifiers with application in bankruptcy prediction**. Technological Forecasting and Social Change, v. 131, p. 111-116, 2018. ISSN 0040-1625.

PECHAC, P. et al. **Implementation of memetic algorithms into structural optimization**. Communications-Scientific letters of the University of Zilina, v. 18, n. 1A, p. 64-69, 2016. ISSN 2585-7878.

PÉTROWSKI, A. **A clearing procedure as a niching method for genetic algorithms**. Proceedings of IEEE international conference on evolutionary computation, 1996, IEEE. p.798-803.

PHAM, D.; KARABOGA, D. **Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks**. Springer Science & Business Media, 2012. ISBN 1447107217.

POMBO, A. V.; MURTA-PINA, J.; PIRES, V. F. **Multiobjective planning of distribution networks incorporating switches and protective devices using a memetic optimization**. Reliability Engineering & System Safety, v. 136, p. 101-108, 2015. ISSN 0951-8320.

POTHIYA, S.; NGAMROO, I.; KONGPRAWECHNON, W. **Application of multiple tabu search algorithm to solve dynamic economic dispatch considering generator constraints**. Energy Conversion and Management, v. 49, n. 4, p. 506-516, 2008. ISSN 0196-8904.

QI, Y. et al. **A memetic multi-objective immune algorithm for reservoir flood control operation**. Water Resources Management, v. 30, n. 9, p. 2957-2977, 2016. ISSN 0920-4741.

RAJASOMASHEKAR, S.; ARAVINDHABABU, P. **Biogeography based optimization technique for best compromise solution of economic emission dispatch**. Swarm and Evolutionary Computation, v. 7, p. 47-57, 2012. ISSN 2210-6502.

RAO, S. S. **Engineering optimization: theory and practice**. John Wiley & Sons, 2009. ISBN 0470183527.

RAO, S. S.; RAO, S. S. **Engineering optimization: theory and practice**. John Wiley & Sons, 2009. ISBN 0470183527.

REYNOLDS, R.; ZANONI, E. **Why cultural evolution can proceed faster than biological evolution**. Proceedings of International Symposium on Simulating Societies, 1992, sn. p.81-93.

REYNOLDS, R. G. **An introduction to cultural algorithms**. Proceedings of the third annual conference on evolutionary programming, 1994, World Scientific. p.131-139.

\_\_\_\_\_. **Cultural algorithms: a tutorial**. Wayne State University, Detroit, 2002.

REYNOLDS, R. G.; ALI, M. **Computing with the social fabric: The evolution of social intelligence within a cultural framework**. IEEE computational intelligence magazine, v. 3, n. 1, p. 18-30, 2008. ISSN 1556-603X.

REYNOLDS, R. G.; CHE, X.; ALI, M. **Weaving the social fabric: The past, present and future of optimization problem solving with cultural algorithms**. International Journal of Intelligent Computing and Cybernetics, v. 3, n. 4, p. 561-592, 2010. ISSN 1756-378X.

REYNOLDS, R. G.; SALEEM, S. **The impact of environmental dynamics on cultural emergence**. Perspectives on Adaptions in Natural and Artificial Systems, p. 253-280, 2005.

REYNOLDS, R. G.; STEFAN, J. M. **Web services, Web searches, and cultural algorithms**. SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483), 2003, IEEE. p.3982-3987.

REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. Editora Manole Ltda, 2003. ISBN 8520416837.

ROCHA, A. M. A.; FERNANDES, E. M. **Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems**. International Journal of Computer Mathematics, v. 86, n. 10-11, p. 1932-1946, 2009. ISSN 0020-7160.

ROSSI-DORIA, O.; PAECHTER, B. **A memetic algorithm for university course timetabling**. Combinatorial optimisation, p. 56, 2004.

SACCO, W. F. et al. **The fuzzy clearing approach for a niching genetic algorithm applied to a nuclear reactor core design optimization problem**. Annals of Nuclear Energy, v. 31, n. 1, p. 55-69, 2004. ISSN 0306-4549.

SALCEDO-SANZ, S.; YAO, X. **Assignment of cells to switches in a cellular mobile network using a hybrid Hopfield network-genetic algorithm approach**. Applied Soft Computing, v. 8, n. 1, p. 216-224, 2008. ISSN 1568-4946.

SAN-JOSÉ-REVUELTA, L. M. **Design of Optimal Frequency-Selective FIR Filters Using a Memetic Algorithm**. 2018 26th European Signal Processing Conference (EUSIPCO), 2018, IEEE. p.1172-1176.

SANTOS, E. B. D. **A ordenação das variáveis no processo de otimização de classificadores bayesianos: Uma abordagem evolutiva**. 2007.

SECUI, D. C. **A new modified artificial bee colony algorithm for the economic dispatch problem.** Energy Conversion and Management, v. 89, p. 43-62, 1/1/ 2015. ISSN 0196-8904. Disponível em: < <http://www.sciencedirect.com/science/article/pii/S0196890414008358> >.

SEGREDO, E.; SEGURA, C.; LEÓN, C. **A multiobjectivised memetic algorithm for the frequency assignment problem.** Evolutionary Computation (CEC), 2011 IEEE Congress on, 2011, IEEE. p.1132-1139.

SHAHOOKAR, K. et al. **Genetic beam search for gate matrix layout.** IEE Proceedings-Computers and Digital Techniques, v. 141, n. 2, p. 123-128, 1994. ISSN 1350-2387.

SHANG, R. et al. **Improved memetic algorithm based on route distance grouping for multiobjective large scale capacitated arc routing problems.** IEEE transactions on cybernetics, v. 46, n. 4, p. 1000-1013, 2016. ISSN 2168-2267.

SHEN, X.-N. et al. **A Q-learning-based memetic algorithm for multi-objective dynamic software project scheduling.** Information Sciences, v. 428, p. 1-29, 2018. ISSN 0020-0255.

SHEKIN, D. J. **Handbook of parametric and nonparametric statistical procedures.** 3rd Ed. CRC Press, 2003. ISBN 1420036262.

SILVA, D. J. A. D. **Algoritmos Culturais com Abordagem Memética e Multipopulacional Aplicados a Problemas de Otimização.** 2012a. 134 f. Doutorado em Engenharia Elétrica (Doutorado). Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Pará, Belém-PA.

\_\_\_\_\_. **Algoritmos culturais com abordagem memética e multipopulacional aplicados a problemas de otimização.** 2012b. (Doutorado em Engenharia Elétrica). Programa de Pós-Graduação em Engenharia Elétrica - PPGEE, Universidade Federal do Pará - UFPA, Belém-PA.

SINHA, N.; CHAKRABARTI, R.; CHATTOPADHYAY, P. **Evolutionary programming techniques for economic load dispatch.** IEEE Transactions on evolutionary computation, v. 7, n. 1, p. 83-94, 2003. ISSN 1089-778X.

SIVANANDAM, S.; DEEPA, S. **Introduction to genetic algorithms.** Springer Science & Business Media, 2007. ISBN 3540731903.

SONKAR, S. et al. **Software Testing using Genetic Algorithm.** International Journal of Computer Science And Technology (IJCST), v. 3, n. 1, p. 183-187, 2012.

SUN, X.-M.; LV, X.-Y.; DUAN, X.-M. **Novel QoS routing algorithm based on cultural-simulated annealing algorithm.** 2009 Second International Conference on Intelligent Networks and Intelligent Systems, 2009, IEEE. p.209-212.

THEODORSSON-NORHEIM, E. **Friedman and Quade tests: BASIC computer program to perform nonparametric two-way analysis of variance and multiple comparisons on ranks of several related samples.** Computers in biology and medicine, v. 17, n. 2, p. 85-99, 1987. ISSN 0010-4825.

TIRRONEN, V. et al. **An enhanced memetic differential evolution in filter design for defect detection in paper production.** *Evolutionary Computation*, v. 16, n. 4, p. 529-555, 2008. ISSN 1063-6560.

UMA, S.; GURUMURTHY, K.; SINGH, M. K. **GA based optimal design of network architecture for desired connectivity and traffic demand.** *International Journal of Scientific & Engineering Research*, v. 2, n. 11, 2011. ISSN 2229-5518.

VEERAMANI, C.; WILLIAMS, J. P.; RAMADEVI, P. **Evaluation of Wind Energy Parameter Optimization of A DFIG Controller Based on Cultural Algorithms.** 2018 International Conference on Communication and Signal Processing (ICCSP), 2018, IEEE. p.957-965.

WANG, J. et al. **Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: formulation, instances, and algorithms.** *IEEE Transactions on Cybernetics*, v. 46, n. 3, p. 582-594, 2016. ISSN 2168-2267.

WANG, Y. et al. **A tabu search based memetic algorithm for the maximum diversity problem.** *Engineering Applications of Artificial Intelligence*, v. 27, p. 103-114, 2014. ISSN 0952-1976.

WARIS, F.; REYNOLDS, R. G. **Optimizing AI Pipelines: A Game-Theoretic Cultural Algorithms Approach.** 2018 IEEE Congress on Evolutionary Computation (CEC), 2018, IEEE. p.1-10.

WEI, Z.; YAN-PING, B.; YE-QING, Z. **The application of an improved cultural algorithm in grid computing.** 2013 25th Chinese Control and Decision Conference (CCDC), 2013, IEEE. p.4565-4570.

WELEKAR, R.; THAKUR, N. V. **An Enhanced Approach to Memetic Algorithm Used for Character Recognition.** In: (Ed.). *Cognitive Informatics and Soft Computing*: Springer, 2019. p.593-602.

WHITLEY, D. et al. **Evaluating evolutionary algorithms.** *Artificial intelligence*, v. 85, n. 1-2, p. 245-276, 1996. ISSN 0004-3702.

WOLPERT, D. H.; MACREADY, W. G. **No free lunch theorems for optimization.** *IEEE transactions on evolutionary computation*, v. 1, n. 1, p. 67-82, 1997. ISSN 1089-778X.

WU, Q.; WANG, Y.; LU, Z. **A tabu search based hybrid evolutionary algorithm for the max-cut problem.** *Applied Soft Computing*, v. 34, p. 827-837, 2015. ISSN 1568-4946.

WU, X.; CHE, A. **A memetic differential evolution algorithm for energy-efficient parallel machine scheduling.** *Omega*, v. 82, p. 155-165, 2019. ISSN 0305-0483.

XHAFA, F. et al. **Efficient batch job scheduling in grids using cellular memetic algorithms.** In: (Ed.). *Metaheuristics for Scheduling in Distributed Computing Environments*: Springer, 2008. p.273-299.

YAN, X. et al. **Cultural algorithm for engineering design problems**. International Journal of Computer Science Issues, v. 9, n. 6, p. 53-61, 2012.

YANG, X.-S.; CHIEN, S. F.; TING, T. O. **Computational intelligence and metaheuristic algorithms with applications**. The Scientific World Journal, v. 2014, 2014. ISSN 2356-6140.

YU, X.; GEN, M. **Introduction to evolutionary algorithms**. Springer Science & Business Media, 2010. ISBN 1849961298.

ZHANG, B.-T.; KIM, J. J. **Comparison of selection methods for evolutionary optimization**. Evolutionary optimization, v. 2, n. 1, p. 55-70, 2000.

ZHANG, Y. **Study on cultural algorithm**. 2011 International Conference on Future Computer Science and Education, 2011, IEEE. p.558-560.

ZHANG, Y. et al. **Memetic algorithm with route decomposing for periodic capacitated arc routing problem**. Applied Soft Computing, v. 52, p. 1130-1142, 2017. ISSN 1568-4946.

ZHOU, Y.; HAO, J.-K.; DUVAL, B. **Opposition-based memetic search for the maximum diversity problem**. IEEE Transactions on Evolutionary Computation, v. 21, n. 5, p. 731-745, 2017. ISSN 1089-778X.

ZHOU, Y.; HAO, J.-K.; GLOVER, F. **Memetic search for identifying critical nodes in sparse graphs**. IEEE transactions on cybernetics, n. 99, p. 1-14, 2018. ISSN 2168-2267.

ZHU, J. **Optimization of power system operation**. John Wiley & Sons, 2015. ISBN 1118854152.

ZHU, Z. et al. **Memetic three-dimensional gabor feature extraction for hyperspectral imagery classification**. International Conference in Swarm Intelligence, 2012, Springer. p.479-488.

ZIANE, I.; BENHAMIDA, F.; GRAA, A. **Simulated annealing optimization for multi-objective economic dispatch solution**. Leonardo Journal of Sciences, n. 25, p. 43-56, 2014. ISSN 1583-0233.

ZUREL, E.; NISAN, N. **An efficient approximate allocation algorithm for combinatorial auctions**. Proceedings of the 3rd ACM conference on Electronic Commerce, 2001, ACM. p.125-136.



## 8 ANEXO

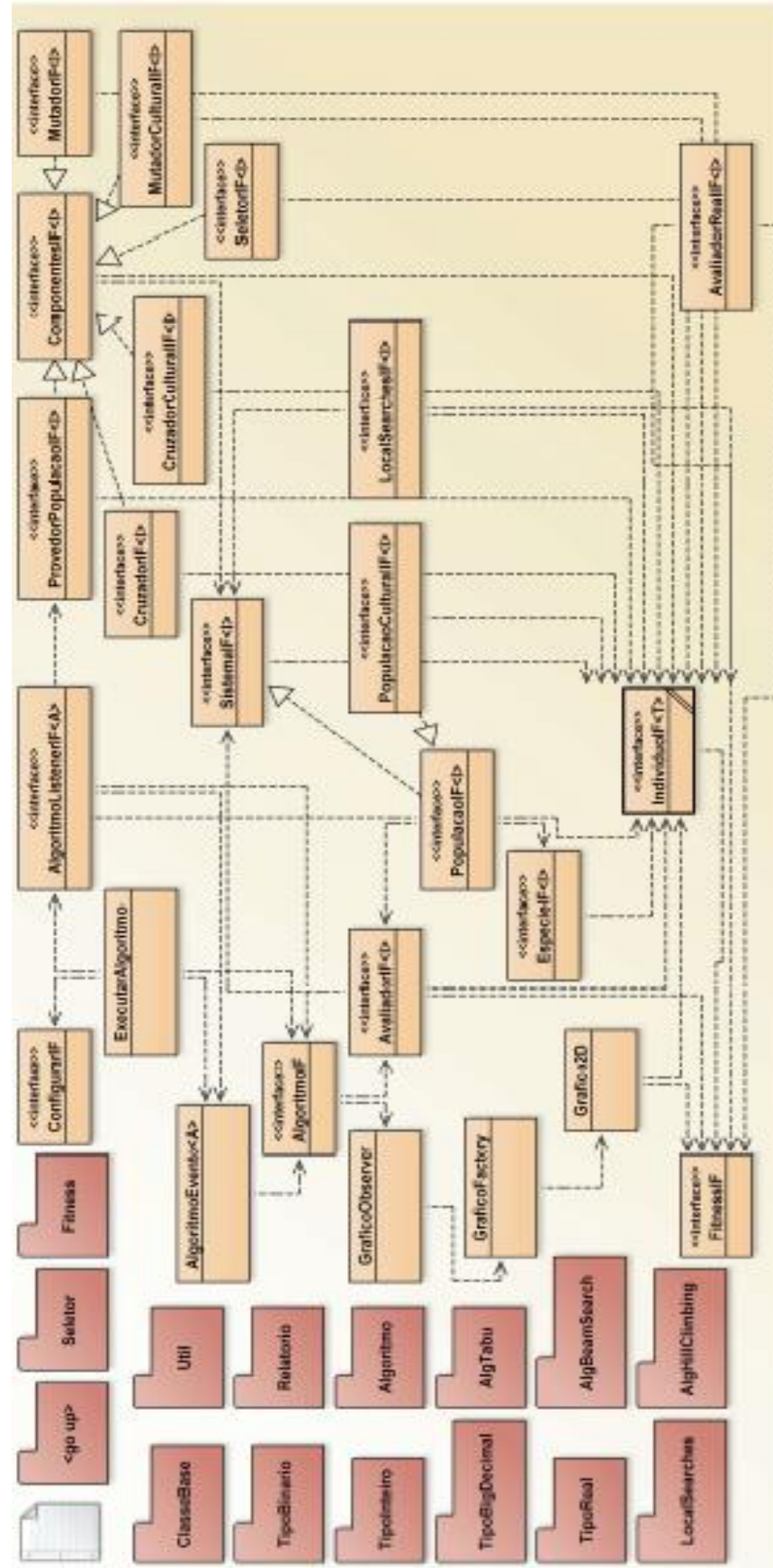
O *Java Evolutionary Framework* – JEF, possibilita o uso de qualquer tipo de algoritmo de Computação Evolutiva (CE). Para tal, se faz necessário o cumprimento de requisitos deste ambiente. A codificação do sistema JEF foi feita em Java, garantindo a portabilidade entre diversas plataformas que implementam uma máquina virtual Java (*Java Virtual Machine* – JVM). A possibilidade de passagem de parâmetros e configurações pelo uso de arquivo no formato XML, fornece a capacidade de variação em testes sem a necessidade de alteração no código fonte e posterior compilação. Mais informações estão disponíveis em: <http://jclec.sourceforge.net/>.

A figura 8.1, mostra a estrutura principal do JEF, já customizado para esta tese.

A figura 8.2, mostra as classes criadas com os modelos dos problemas utilizados nesta tese.

A figura 8.3, mostra o código fonte da função *Bent Cigar* – F1.

Figura 8.1: Estructura principal do JEF Customizada.



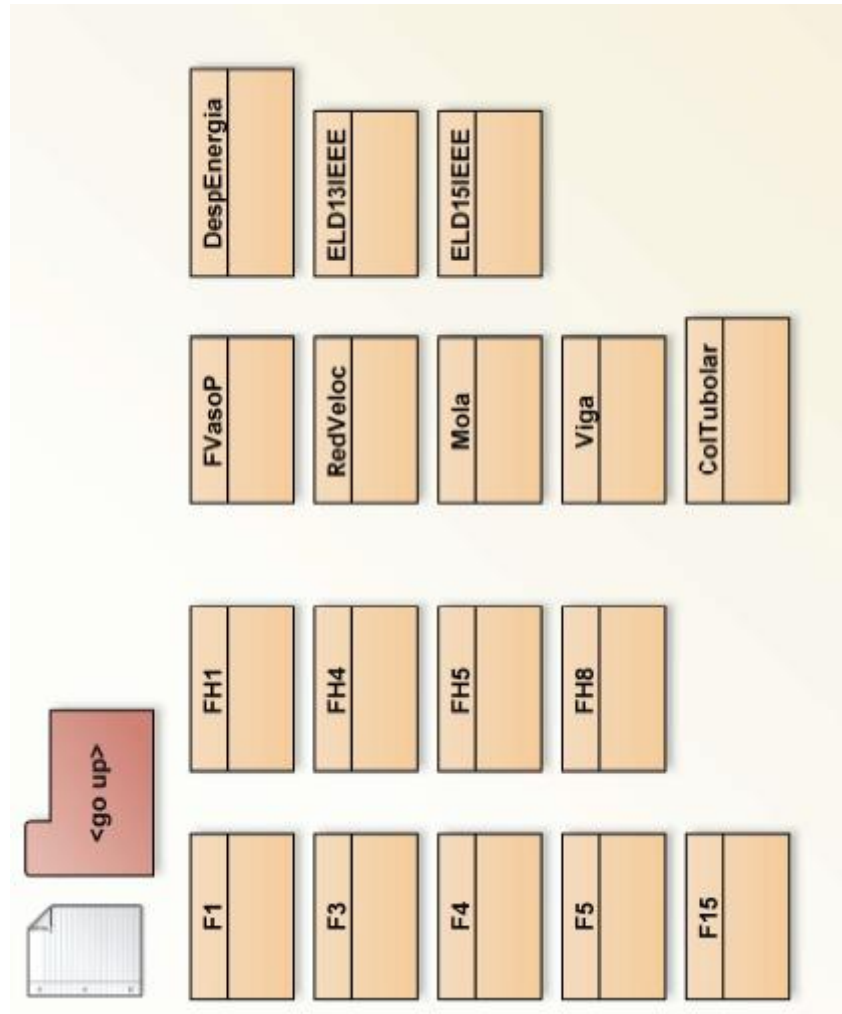
**Figura 8.2:** Classes de modelagem dos problemas.

Figura 8.3: Código fonte da função F1.

```

Class F1 - FEJ_v6_0 1/2
1 package jef.TipoReal.GrupoBenchmark2;
2 import jef.ClasseBase.AvaliadorAbs;
3 import jef.TipoReal.IndividuoReal;
4 import jef.Util.Minimiser.usaMinimo;
5 import java.math.*;
6 import jef.FitnessIF;
7 import jef.EspecieIF;
8 import jef.Fitness.FitnessValorSimples;
9 import jef.Fitness.FitnessValorComparator;
10 import java.util.List;
11 import java.util.ArrayList;
12 import java.util.Comparator;
13 /**
14  * Correspondente Bent Cigar Function = F1 (CEC2017)
15  * @author (Carlos Freitas) @version (18-Des-2016)
16  */
17 public class F1 extends AvaliadorAbs<IndividuoReal>
18 {
19     private double x,y,z;
20     private static double xm,ym;
21
22     public void avaliar(IndividuoReal indiv)
23     {
24         double fitness= calculaAvaliacao_F1(indiv);
25         indiv.setarFitness(new FitnessValorSimples(fitness));
26     }
27     public double avaliar(IndividuoReal indiv,int x)
28     {
29         double fitness= calculaAvaliacao_F1(indiv);
30         return fitness;
31     }
32     public List pegarParametrosDoAvaliador()
33     {
34         return (null);
35     }
36     /** n=30 [-100.0:100.0] minimo->x=?;f(x)=0.0 Correspondente Bent Cigar Fun
37     ction = F1 (CEC2017) */
38     public double calculaAvaliacao_F1(IndividuoReal indiv)
39     {
40         double [] cromosomaReal=indiv.pegarCromossomo();
41         indiv.setaVariaveis(cromosomaReal);
42         y=cromosomaReal[0];
43         double soma=0.0;
44         double somaf=0.0;
45         for (int i=1;i<this.dimensao;i++)
46         {
47             x=cromosomaReal[i];
48             soma=soma+Math.pow(x, 2);
49         }
50         somaf=Math.pow(y,2)+ Math.pow(10,6)*soma;
51         double ResAvaliacao=somaf;
52         return (ResAvaliacao);
53     }
54     public Comparator<FitnessIF> pegaComparador()
55     {
56         return new FitnessValorComparator();
57     }
58 }

```