



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

DESENVOLVIMENTO DE UM ALGORITMO MULTIOBJETIVO HÍBRIDO: ESTUDO DE CASO NA GERAÇÃO DE ENERGIA COM USINAS HIDROELÉTRICAS EM CASCATA

LUCAS BRAGA DE OLIVEIRA

Orientador: Paulo Eduardo Maciel de Almeida
Centro Federal de Educação Tecnológica de Minas Gerais

Coorientadora: Carolina Gil Marcelino
Universidade Federal do Rio de Janeiro

BELO HORIZONTE
FEVEREIRO DE 2019

LUCAS BRAGA DE OLIVEIRA

DESENVOLVIMENTO DE UM ALGORITMO MULTIOBJETIVO HÍBRIDO: ESTUDO DE CASO NA GERAÇÃO DE ENERGIA COM USINAS HIDROELÉTRICAS EM CASCATA

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Mestre em Modelagem Matemática e Computacional.

Área de concentração: Modelagem Matemática e Computacional

Linha de pesquisa: Sistemas Inteligentes

Orientador: Paulo Eduardo Maciel de Almeida
Centro Federal de Educação Tecnológica de Minas Gerais

Coorientadora: Carolina Gil Marcelino
Universidade Federal do Rio de Janeiro

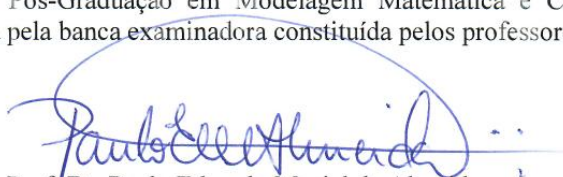
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL
BELO HORIZONTE
FEVEREIRO DE 2019



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

**“DESENVOLVIMENTO DE UM ALGORITMO MULTI OBJETIVO
HÍBRIDO: ESTUDO DE CASO NA GERAÇÃO DE ENERGIA COM
USINAS HIDROELÉTRICAS EM CASCATA”**

Dissertação de Mestrado apresentada por **Lucas Braga de Oliveira**, em 26 de fevereiro de 2019, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:


Prof. Dr. Paulo Eduardo Maciel de Almeida
Centro Federal de Educação Tecnológica de Minas Gerais


Prof^a. Dr^a. Carolina Gil Marcelino
Centro Federal de Educação Tecnológica de Minas Gerais


Prof. Dr. Rodney Rezende Saldanha
Universidade Federal de Minas Gerais


Prof^a. Dr^a. Elizabeth Fialho Wanner
Centro Federal de Educação Tecnológica de Minas Gerais

Visto e permitida a impressão,


Prof. Dr. Thiago de Souza Rodrigues
Coordenador do Programa de Pós-Graduação em
Modelagem Matemática e Computacional

Agradecimentos

Aos meus orientadores Paulo e Carol, pela orientação e paciência durante a realização deste trabalho.

À professora Hersília, pela ajuda ao esclarecer minhas dúvidas sobre balanço hídrico e por fornecer o contato necessário para obter os dados utilizados neste trabalho.

Ao engenheiro Ivan, por fornecer dados e informações imprescindíveis para a realização deste trabalho

À banca, pelas sugestões que colaboraram para a melhoria deste trabalho e de futuras publicações. Aos colegas de laboratório, Wagner, Túlio, Rondy e Ramon pela companhia, amizade, ajuda e disponibilidade durante o mestrado.

Aos meus pais e minha irmã, pelo apoio e carinho durante os tempos difíceis.

Aos amigos, pelo companheirismo, apoio e amizade.

À todos, Muito obrigado.

Resumo

Os algoritmos evolucionários se inspiram em mecanismos de adaptação encontrados na natureza e nas últimas décadas têm se destacado como solução para diversos problemas nas áreas de engenharia, ciência e médica. À medida que a tecnologia avança, a complexidade de seus sistemas aumenta de forma que se torna árduo a sua solução sem o auxílio computacional destes algoritmos. Uma das técnicas para se melhorar o desempenho destes algoritmos é a hibridização, que consiste na incorporação e combinação de diferentes algoritmos e técnicas de forma que as vantagens de um algoritmo ou técnica suprimam as deficiências do outro, melhorando de maneira geral a performance. Neste trabalho, foi desenvolvido o algoritmo multiobjetivo híbrido MESH, que se fundamenta nos paradigmas evolucionário e de busca por enxame do algoritmo híbrido C-DEEPSO e incorpora técnicas de outros algoritmos multiobjetivo. O MESH foi comparado com os algoritmos multiobjetivo clássicos NSGA-II e SPEA2, nas funções *benchmark* ZDT e em dois modelos multiobjetivo de geração de energia em usinas hidroelétricas. Um dos modelos avaliados, que é proposto neste trabalho, aborda a dinâmica da geração de energia com usinas em cascata com os objetivos de eficiência energética e manutenção dos níveis dos reservatórios do sistema. Em problemas *benchmark* o MESH se mostrou competitivo em relação aos dois algoritmos clássicos, obtendo soluções de forma computacionalmente mais rápida e de qualidade igual ou superior. Nas avaliações com problemas reais de geração de energia com usinas hidroelétricas, o MESH se apresentou sensível à combinação de sua configuração de parâmetros e do tipo de problema a ser otimizado. Das duas configurações avaliadas, uma foi capaz de obter soluções de qualidade superior ao NSGA-II e SPEA2, enquanto a outra obteve soluções inferiores. O modelo matemático proposto por sua vez demonstrou que possível ser mais eficiente que o modo de operação padrão adotado em usinas hidrelétricas brasileiras, podendo economizar grandes quantias de água.

Palavras-chave: Inteligência Computacional; Algoritmos Evolucionários; Algoritmos Híbridos; Otimização Multiobjetivo, Hidroelétricas em cascata.

Lista de Figuras

Figura 1 – Balanço Energético de acordo com a fonte do ano de 2018.	2
Figura 2 – Fronteira Pareto-ótimo em um espaço objetivo bidimensional.	10
Figura 3 – Funcionamento do DE.	12
Figura 4 – Contribuição dos termos na equação de movimento do PSO.	15
Figura 5 – Etapas de funcionamento de um algoritmo genético típico.	20
Figura 6 – Fronteiras de soluções no espaço objetivo.	21
Figura 7 – Principais etapas de uma iteração do NSGA-II.	24
Figura 8 – Operador de truncamento do SPEA2, a esquerda pareto original e a direita o pareto destacando os indivíduos a serem removidos para um truncamento para $\overline{N} = 5$	25
Figura 9 – Procedimento de escolha para os melhores globais no MEPSO. As fronteiras da população escolhem os melhores globais baseando-se na fronteira superior seguinte. A primeira fronteira por sua vez escolhe os melhores globais a partir da lista pareto contendo as soluções não dominadas encontradas pelo algoritmo.	29
Figura 10 – Corte lateral de uma UHE e seus os principais componentes.	30
Figura 11 – Principais variáveis e cotas envolvidas no balanço hídrico do reservatório de uma UHE.	32
Figura 12 – Pareto ótimo para a ZDT1.	41
Figura 13 – Pareto ótimo para a ZDT2.	42
Figura 14 – Pareto ótimo para a ZDT3.	43
Figura 15 – Pareto ótimo para a ZDT4.	44
Figura 16 – Pareto ótimo para a ZDT6.	45
Figura 17 – Diagrama do funcionamento do MESH.	49
Figura 18 – Diagrama para a escolha do guia de enxame de uma partícula no MESH. No método apresentado à esquerda, todas partículas da população escolhem a partir da memória. No método apresentado à direita, a escolha é feita baseando-se na fronteira superior seguinte em relação à fronteira que a partícula pertence. As partículas da primeira fronteira por sua vez utilizam da memória.	50
Figura 19 – Diagrama do funcionamento do operador de mutação diferencial do MESH.	50
Figura 20 – Pareto combinado dos algoritmos nas funções ZDT.	62
Figura 21 – <i>Boxplot</i> dos hipervolumes dos três algoritmos na função ZDT1.	63
Figura 22 – Teste de Tukey dos hipervolumes dos três algoritmos na função ZDT1.	64

Figura 23 – Pareto combinado dos algoritmos com o modelo de distância entre os controles otimizado e conjunto. Os números indicam o tamanho dos paretos dos algoritmos.	68
Figura 24 – <i>Boxplot</i> dos hipervolumes dos três algoritmos no modelo de distância entre os controles otimizado e conjunto.	69
Figura 25 – Teste de Tukey dos hipervolumes dos três algoritmos no modelo de distância entre os controles otimizado e conjunto.	69
Figura 26 – Diagrama para a simulação de uma iteração do modelo.	70
Figura 27 – Diagrama do sistema de UHE em cascata usado na simulação.	71
Figura 28 – Pareto combinado dos algoritmos no modelo de cascata nas horas 0-5.	78
Figura 29 – Pareto combinado dos algoritmos no modelo de cascata nas horas 6-11.	79
Figura 30 – Pareto combinado dos algoritmos no modelo de cascata nas horas 12-17.	80
Figura 31 – Pareto combinado dos algoritmos no modelo de cascata nas horas 18-23.	81
Figura 32 – <i>Boxplot</i> dos hipervolumes dos três algoritmos no modelo de cascata para a primeira hora de simulação.	82
Figura 33 – Teste de Tukey dos hipervolumes dos três algoritmos no modelo de cascata para a primeira hora de simulação.	83

Lista de Tabelas

Tabela 1 – Coeficientes Operativos	33
Tabela 2 – Relação índice dos gráficos com as configurações de MESH	58
Tabela 3 – Síntese das análises dos resultados do experimento 1.	59
Tabela 4 – Compilação das melhores configurações do MESH para a ZDT6, ZDT4 e ZDT3	60
Tabela 5 – Análise dos hipervolumes dos algoritmos nas funções ZDT.	63
Tabela 6 – Análise das distâncias geracionais dos algoritmos nas funções ZDT.	65
Tabela 7 – Média da diferença entre os hipervolumes encontrados e o hipervolume máximo no experimento 2.	65
Tabela 8 – Média das distâncias geracionais do experimento 2.	66
Tabela 9 – Coeficientes dos polinômios das cotas montante e jusante.	72
Tabela 10 – Vazões afluentes para a simulação	73
Tabela 11 – Demandas de energia para a simulação.	73
Tabela 12 – Valores das funções objetivo das soluções escolhidas, no qual: F_1 - Eficiência energética média do sistema em MW por $m^3 s^{-1}$ F_2 - Nível médio dos reservatórios do sistema em %. Soluções em verde dominam o controle conjunto. Soluções em laranja são dominadas pelo controle conjunto.	75
Tabela 13 – Valores das funções objetivo das soluções escolhidas, no qual: F_1 - Eficiência energética média do sistema em MW por $m^3 s^{-1}$ F_2 - Nível médio dos reservatórios do sistema em %. Soluções em verde dominam o controle conjunto. Soluções em laranja são dominadas pelo controle conjunto.	76
Tabela 14 – Análise dos hipervolumes dos algoritmos no modelo de cascata.	83

Lista de Quadros

Quadro 1 – Descrição dos parâmetros utilizados no modelo.	34
---	----

Lista de Algoritmos

Algoritmo 1 – Pseudo-código do funcionamento do PSO	14
Algoritmo 2 – Pseudo-código do funcionamento do C-DEEPSO	18
Algoritmo 3 – Pseudo-código para o cálculo das variáveis n_p , s_p , p_{rank} e da criação da primeira fronteira	22
Algoritmo 4 – Pseudo-código para criação das demais fronteiras	22
Algoritmo 5 – Pseudo-código para o cálculo da distância de multidão	23

Sumário

1 – Introdução	1
1.1 Contexto	1
1.2 Revisão bibliográfica	3
1.2.1 Otimização de usinas hidroelétricas	3
1.2.2 Algoritmos evolutivos	5
1.3 Objetivos	6
1.3.1 Objetivo geral	6
1.3.2 Objetivos específicos	7
1.4 Organização do trabalho	7
2 – Fundamentação teórica	8
2.1 Otimização mono-objetivo	8
2.1.1 Otimização multiobjetivo	9
2.1.1.1 Dominância entre soluções e pareto-ótimo	9
2.1.2 Otimização dinâmica	10
2.2 Computação evolutiva	11
2.2.1 Algoritmos mono-objetivo	12
2.2.1.1 Evolução diferencial (DE)	12
2.2.1.2 Otimização por enxame de partículas (PSO)	14
2.2.1.3 Otimização por enxame de partículas evolucionário (EPSO)	15
2.2.1.4 <i>Differential evolutionary particle swarm optimization</i> (DE-EPSO)	16
2.2.1.5 <i>Canonical differential evolutionary particle swarm optimization</i> (C-DEEPSO)	17
2.2.2 Algoritmos multiobjetivo	19
2.2.2.1 <i>Nondominated sorting genetic algorithm II</i> (NSGA-II)	19
2.2.2.2 <i>Strength pareto evolutionary algorithm 2</i> (SPEA2)	23
2.2.2.3 Abordagens multiobjetivo para otimização por enxame de partículas	26
2.2.2.4 Abordagem multi-objetivo para evolução diferencial	27
2.2.2.5 Multi-objective evolutionary particle swarm optimization (MEPSO)	28
2.3 Usina hidroelétrica	29
2.3.0.1 Principais componentes de uma UHE	30
2.3.1 Balanço hídrico de um reservatório de uma UHE	31
2.3.2 Modelo matemático para o despacho elétrico de uma UHE	33

3 – Metodologia	36
3.1 Planejamento de experimentos	37
3.1.1 Experimentos planejados	37
3.1.2 Métricas de comparação de algoritmos multiobjetivo	38
3.1.3 Métodos para análise estatística	39
3.1.4 Pareto combinado	40
3.1.5 Funções <i>benchmark</i>	41
3.1.5.1 ZDT1	41
3.1.5.2 ZDT2	42
3.1.5.3 ZDT3	43
3.1.5.4 ZDT4	44
3.1.5.5 ZDT6	45
4 – Desenvolvimento do algoritmo híbrido multiobjetivo MESH	46
4.1 <i>Multi-objective evolutionary swarm hybrid</i> (MESH)	46
4.2 Considerações finais	51
5 – Proposta de modelo matemático para a geração de energia em UHE em cas-	
cata	52
6 – Experimentos e análise dos resultados	57
6.1 Ambiente e parâmetros dos experimentos	57
6.2 Experimento 1: validação das configurações do MESH	57
6.3 Experimento 2: avaliação de desempenho em funções <i>benchmark</i>	61
6.3.1 Paretos combinados	61
6.3.2 Análise das métricas de comparação	63
6.3.3 Considerações finais	65
6.4 Experimento 3: avaliação em problemas reais e validação do modelo de UHE em cascata proposto	67
6.4.1 Avaliação para o modelo de distância entre os controles otimizado e conjunto	67
6.4.2 Estrutura para simulação do modelo de cascata proposto	70
6.4.3 Descrição do sistema de usinas em cascata e suas variáveis para simulação	71
6.4.4 Avaliação das soluções para o modelo de cascata	74
6.4.5 Avaliação do desempenho dos algoritmos no modelo de cascata	77
6.4.6 Considerações finais	84
7 – Conclusão e trabalhos futuros	85
7.1 Conclusão	85
7.2 Trabalhos futuros	86

Referências 88

1 Introdução

1.1 Contexto

A inteligência computacional é uma área de pesquisa da computação que possui sistemas de métodos e infraestrutura com o propósito de melhorar a inteligência humana, aprendendo e descobrindo padrões, relações e estruturas em ambientes dinâmicos e complexos, para resolver problemas práticos (KORDON, 2009). Os paradigmas centrais da inteligência computacional envolvem: as redes neurais artificiais, sistemas nebulosos e computação evolutiva. A computação evolutiva é um dos paradigmas que tem se destacado nas últimas décadas para solução de diversos problemas reais da engenharia e ciência, pois à medida que a tecnologia avança, muitos sistemas possuem modelos cada vez mais complexos e que permitem um grande número de soluções, o que torna virtualmente inviável a realização de um projeto sem o auxílio direto ou indireto de técnicas de computação evolutiva (LOPES; TAKAHASHI, 2011).

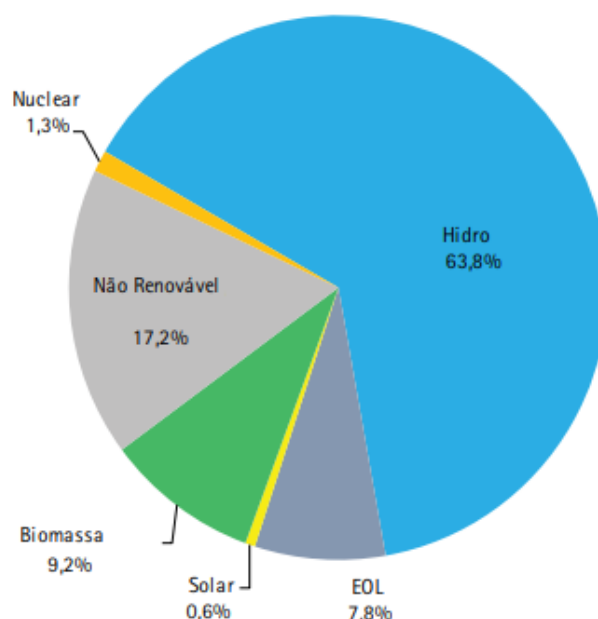
As soluções de computação evolutiva são meta-heurísticas que se inspiram nos mecanismos de adaptação biológica presentes na natureza. Diversos algoritmos foram desenvolvidos e inspirados em mecanismos bio-inspirados como: a teoria da evolução das espécies, a dinâmica de uma colônia de formigas à procura de alimento, a movimentação de enxames. Uma alternativa para se melhorar o desempenho de algoritmos evolutivos é a hibridização, que consiste na combinação de mecanismos de diferentes algoritmos de forma que as desvantagens de um algoritmo são suprimidas pelas vantagens de outro.

A área de geração de energia elétrica é uma das que se beneficiam do uso da computação evolutiva. Isso ocorre ao aplicar por exemplo, algoritmos para a otimização de modelos de sistemas de geração e transmissão de energia, de forma que se encontre novas maneiras mais eficientes para se gerar energia. Neste trabalho um tema abordado é a geração de energia elétrica por usinas hidroelétricas, que será utilizado como estudo de caso a ser otimizado pelo algoritmo híbrido de computação evolutiva desenvolvido neste trabalho.

Conforme apresentado na Figura 1, o sistema de geração de energia do Brasil é majoritariamente composto de fontes renováveis, sendo 63,8% de fontes hidráulicas. Por causa disso, usinas hidroelétricas (UHE) de diferentes portes estão presentes no território nacional. As UHE operam de forma a serem capazes de atender as demandas de energia determinadas pelo Operador Nacional de Sistema Elétrico (ONS), órgão responsável pela coordenação e controle da operação das instalações de geração e transmissão de energia elétrica no Sistema Interligado Nacional (SIN) e pelo planejamento da operação

dos sistemas isolados do país, sob a fiscalização e regulação da Agência Nacional de Energia Elétrica (Aneel) (ONS, 2018). Apesar das fontes hidráulicas comporem a maior parte do balanço energético do Brasil, o seu percentual tem caído nos últimos anos devido a condições hidrológicas desfavoráveis. Por causa disto, passa a ser cada vez mais necessário o manejo otimizado dos recursos hídricos disponíveis, por meio do uso de algoritmos evolucionários híbridos.

Figura 1 – Balanço Energético de acordo com a fonte do ano de 2018.



Fonte: (ONS, 2018).

Grande parte das UHE brasileiras possuem um sistema automatizado de controle de geração de energia, geralmente chamado de controle conjunto. Esses sistemas, ao receberem uma demanda de energia a ser gerada via ONS, distribuem de maneira igualitária o despacho de energia para cada unidade composta pelo conjunto turbina-gerador. Trabalhos relacionados mostram que essa divisão igualitária não é eficiente sobre diversos aspectos como eficiência energética ou economia de água e muitos destes trabalhos utilizam algoritmos de computação evolutiva para encontrar soluções para seus modelos matemáticos.

Para tratar o estudo de caso sobre a geração de energia de UHE, foram utilizados o modelo de Marcelino (2012) e Marcelino (2017) como ponto de partida para este trabalho visando a proposição e desenvolvimento de um novo modelo. O modelo presente em Marcelino (2012) e Marcelino, Wanner e Almeida (2013) é mono-objetivo e trata o problema de despacho elétrico de uma UHE. Tal modelo pode ser definido como a atribuição de valores operativos para cada conjunto turbina-gerador de uma usina, a partir de critérios como demanda de energia a ser produzida, limites operativos dos conjuntos e outros.

Uma abordagem multiobjetivo foi proposta por [Marcelino et al. \(2015\)](#); [Marcelino \(2017\)](#). Na qual dois objetivos foram tratados: maximização da eficiência da produção de energia e a minimização da diferença do uso de água em relação ao modo controle conjunto usual. O modelo desenvolvido neste trabalho é um modelo multiobjetivo e que trata da dinâmica de geração de energia quando um conjunto de usinas estão em cascata.

Este trabalho apresenta um aprimoramento dos modelos matemáticos de despacho elétrico predecessores, que trata o problema de geração de energia elétrica para um conjunto de usinas dispostas em cascata. Este novo modelo possui abordagem multiobjetivo tratando da eficiência energética das UHEs e da manutenção dos níveis dos reservatórios.

1.2 Revisão bibliográfica

A revisão bibliográfica realizada pode ser dividida em dois temas: otimização de modelos de geração de energia por usinas hidroelétricas; algoritmos multiobjetivos e híbridos relacionados aos conceitos utilizados no desenvolvimento do algoritmo proposto por [Marcelino et al. \(2018\)](#).

1.2.1 Otimização de usinas hidroelétricas

Este trabalho está relacionado diretamente com o trabalho de [Marcelino \(2017\)](#), visto que propôs novas abordagens para o algoritmo híbrido e para o modelo de geração de energia elétrica de UHE. Em seu trabalho, [Marcelino \(2017\)](#) contribuiu para a melhoria da eficiência energética por meio do desenvolvimento de dois algoritmos híbridos. Os algoritmos solucionaram diversos problemas ligados ao setor energético, tais como: problemas de fluxo de potência ótimo (OPF) no despacho elétrico de UHE e usinas eólicas, despacho elétrico em redes *microgrid* e problemas de fluxo de potência com restrições de segurança (SCOPF) no despacho ativo/reactivo em redes de larga escala.

Para tratar o despacho elétrico em UHE, [Marcelino \(2017\)](#) deu continuidade ao modelo no trabalho de [Marcelino \(2012\)](#). Este modelo tem como objetivo a maximização da eficiência energética na produção de energia da UHE, que é dada pela razão do somatório da potência produzida por cada unidade geradora, dividida pelo somatório das vazões de água em cada unidade geradora. Desta forma maximizar esta função significa que a UHE gera uma determinada potência utilizando uma menor vazão de água. Este modelo destaca-se por tratar questões como as perdas hidráulicas, a altura da queda bruta e a ideia de diferenciar cada unidade-geradora em vez de representá-las como unidades idênticas, comumente estes fatores são negligenciados ou simplificados em outros modelos da literatura. Os resultados demonstram a capacidade do modelo de aperfeiçoar a eficiência energética da UHE, sendo capaz de economizar até 15,7 milhões de m^3 de água por mês.

A tese de [Fernandes \(2015\)](#) apresenta uma metodologia para otimização da operação de UHE, que se baseia na programação por metas lexicográficas utilizando o algoritmo genético mono-objetivo e o *strength pareto evolutionary algorithm*. O modelo matemático possui dois objetivos, sendo que o primeiro é a maximização da geração líquida da usina ao longo de um dia. O segundo objetivo é a minimização da quantidade de partidas e paradas das unidades geradoras. [Fernandes \(2015\)](#) denominam este tipo de problema como despacho dinâmico de máquinas e geração. A resolução é executada em duas etapas, na primeira etapa apenas o algoritmo genético é aplicado para resolver o problema estático para cada hora; na segunda etapa ambos algoritmos são empregados para solucionar o problema dinâmico ao longo de um dia. As UHE Jupia e Porto Primavera são os estudos de casos deste trabalho, fatores como perdas hidráulicas e elétricas não são consideradas no modelo.

[Lisboa, Guedes e Vieira \(2013\)](#) discutem modelos multiobjetivos para otimização da operação de UHE em cascata. As funções objetivo modeladas são a maximização da geração e do volume final dos reservatórios. O segundo objetivo está relacionado com a garantia da geração futura. O primeiro objetivo é tratado como linear, com produtividade constante e não linear, com produtividade variando de acordo com o nível do reservatório; com isso o problema é modelado pela programação linear e não linear. Os resultados do trabalho demonstram que o modelo não linear domina o modelo linear.

[Scuzziato \(2011\)](#) em sua dissertação propõe um modelo para a alocação de unidades hidrelétricas, com objetivo de determinar quais unidades devem operar e os respectivos níveis de geração de UHE em cascata, em uma janela de tempo de um dia. Fatores de perdas hidráulicas, elétricas e mecânicas são considerados o que o torna um dos mais completos modelos na literatura. Para solucionar o modelo, uma estratégia de decomposição baseada nos métodos da Relaxação Lagrangeana e do Lagrangeano Aumentado foi proposta. A solução é aplicada a um estudo de caso que consiste em um sistema composto por quatro UHE em cascata, cuja a capacidade de potência instalada é de 4170 MW. O modelo proposto por [Scuzziato \(2011\)](#) é mono-objetivo, porém um dos termos desta função objetivo é um somatório que contabiliza quando uma unidade geradora muda de estado (ligada ou desligada). Este somatório penaliza a função objetivo de maneira semelhante ao segundo objetivo de [Fernandes \(2015\)](#), portanto o modelo poderia ser adaptado para se tornar multiobjetivo.

O trabalho de [Li e Qiu \(2015\)](#) propõe um modelo de otimização dos reservatórios de UHE em cascata, com os objetivos de maximizar a geração de energia e a potência firme. As variáveis de decisão são a vazão de água nos reservatórios e o planejamento anual. O algoritmo *Nondominated Sorting Genetic Algorithm II* foi utilizado para solucionar o problema, cujo estudo de caso foi o sistema das Três Gargantas localizado na China.

1.2.2 Algoritmos evolutivos

O algoritmo híbrido *Canonical Differential Evolutionary Particle Swarm Optimization* (C-DEEPSO) incorpora os conceitos de teoria evolutiva e inteligência de enxames, respectivamente dos algoritmos evolução diferencial e otimização por enxame de partículas. Nos problemas de despacho de energia para UHE, otimização de um sistema *microgrid* híbrido e no sistema IEEE de 41 barras para uma usina eólica instalado no mar, o C-DEEPSO apresentou um desempenho superior a outros algoritmos da literatura. Para solução dos sistemas IEEE 57, 118 e 300 barras o algoritmo proposto foi o *Hybrid Canonical Differential Evolutionary Particle Swarm Optimization* (hC-DEEPSO). Devido à grande quantidade de variáveis de decisão e da alta complexidade destes problemas, foi acrescentado ao término de cada iteração do C-DEEPSO uma busca local inspirada na geração de espirais logarítmicas. Além disto, um mecanismo para manter a diversidade de soluções, criando assim o hC-DEEPSO (MARCELINO, 2017; MARCELINO et al., 2018).

O algoritmo de evolução diferencial proposto por Storn e Price (1997) é uma meta-heurística que mantém um conjunto de soluções denominado de população. Esta população de soluções passa por processos de mutação, recombinação e seleção, gerando novas soluções à cada iteração de seu funcionamento. Sua principal característica se encontra na maneira em que de novas soluções são geradas. Isto é realizado, em sua forma canônica, pela diferença vetorial de duas soluções adicionada ao vetor de uma terceira solução.

Inspirados no comportamento social de animais quando se deslocam em conjuntos, Kennedy e Eberhart (1995) propuseram o algoritmo de otimização por enxame de partículas (PSO), no qual as soluções são denominadas partículas. Estas se deslocam seguindo uma equação de movimento que possui três componentes: a inércia que mantém a sua velocidade, a memória que possui informações do melhor resultado já encontrado e a cooperação, que semelhante à memória mantém, o melhor resultado encontrado por todo enxame.

Miranda e Fonseca (2002), ao combinar os conceitos da estratégia evolutiva com o PSO, desenvolveram a otimização por enxame de partículas evolucionário (EPSO). Os elementos evolutivos tornam o algoritmo auto adaptativo e tiram a responsabilidade do usuário de escolher os parâmetros oriundos do PSO que tem impacto direto na performance do algoritmo.

Miranda e Alves (2013) aprimoraram o EPSO pela adição de partes do algoritmo de evolução diferencial na equação de movimento do EPSO, criando o Differential Evolutionary Particle Swarm Optimization (DEEPSO). Comparando-se com a equação do EPSO, no DEEPSO o termo relativo à memória é substituído pelo vetor de diferenças e o termo de cooperação possui uma nova matriz P , que permite que a contribuição do termo de cooperação seja parcial. Miranda e Alves (2013) também criaram quatro estratégias para a

escolha das partículas envolvidas no vetor de diferenças.

O *Nondominated Sorting Genetic Algorithm II* (NSGA-II) de [Deb \(2002\)](#) é um algoritmo genético capaz de resolver problemas multiobjetivos. Os algoritmos genéticos são inspirados nos mecanismos de seleção natural e operam sobre uma população de soluções, no qual os processos de mutação, seleção e recombinação são aplicados à cada iteração gerando novas soluções. Estes algoritmos foram desenvolvidos inicialmente para resolução de problemas mono-objetivo, para tratar problemas multiobjetivo [Deb \(2002\)](#) propôs dois novos processos com o NSGA-II, um deles separa a população em fronteiras baseando-se no conceito de dominância e outro quantifica a distância entre os membros de uma mesma fronteira. Com esses dois processos, a dinâmica do algoritmo genético pode ser aplicado em problemas multiobjetivo.

Assim como o NSGA-II, o *Strength Pareto Evolutionary Algorithm 2* (SPEA2) de [Zitzler, Laumanns e Thiele \(2001\)](#) trata problemas multiobjetivos com técnicas baseadas nos conceitos de dominância, porém suas soluções ótimas são mantidas em um arquivo externo à população. Apesar de utilizar o conceito de dominância entre as soluções, a população não é separada em fronteiras como no NSGA-II e o estimador de densidade dos membros não dominados é realizado pelo método de k-vizinhos mais próximos.

O multi-objective evolutionary particle swarm optimization (MEPSO) de [Maciel et al. \(2012\)](#) é uma abordagem multiobjetivo do algoritmo EPSO. O MEPSO modifica a equação de movimento e o processo de seleção das melhores partículas para a iteração seguinte, de forma que as decisões envolvidas nestas duas etapas utilizem os resultados dos dois procedimentos do NSGA-II.

No âmbito de estratégias multiobjetivo para os algoritmos de evolução diferencial (DE) e otimização por enxame de partículas (PSO), temos o trabalho de [Xue, Sanderson e Graves \(2003\)](#) que apresenta uma versão multiobjetivo para o DE, o trabalho de [Coello e Lechuga \(2002\)](#) para uma abordagem multiobjetivo para o PSO e o trabalho de [Mostaghim e Teich \(2003\)](#) que discute diferentes maneiras de se escolher as partículas para o termo de cooperação em um contexto multiobjetivo. Maiores detalhes desses trabalhos serão apresentados na seção de fundamentação teórica.

1.3 Objetivos

1.3.1 Objetivo geral

Este trabalho tem como objetivo geral contribuir para a área da computação evolutiva, com um novo algoritmo que segue o paradigma de hibridização para o contexto de otimização de problemas multiobjetivo. O algoritmo será avaliado em problemas de teste e em problemas reais com estudos de casos na área de energia elétrica em UHE, no qual

também tem como objetivo contribuir com um novo modelo de geração de energia com usinas em cascata.

1.3.2 Objetivos específicos

O presente trabalho possui como objetivos específicos:

1. Desenvolver um novo algoritmo híbrido multiobjetivo utilizando como base o funcionamento do C-DEEPSO e incorporando mecanismos e ideias de algoritmos MO na área de computação evolutiva.
2. Validar o desempenho do algoritmo proposto em funções *benchmark* comparando-o com outros algoritmos multiobjetivo consolidados na literatura.
3. Propor um novo modelo multiobjetivo para a geração de energia de UHE em cascata, utilizando o modelo de despacho elétrico de [Marcelino \(2017\)](#), como modelo de geração de energia UHE.
4. Validar o desempenho do algoritmo proposto para problemas reais, o algoritmo será aplicado sobre os modelos de despacho elétrico da energia em UHE, sendo que um deles é o modelo proposto no objetivo anterior.

1.4 Organização do trabalho

Este trabalho está organizado em sete capítulos, sendo o primeiro deles a introdução. O segundo capítulo apresenta a fundamentação teórica necessária para o entendimento e realização deste trabalho, incluindo conceitos referentes à otimização mono-objetivo e multiobjetivo, algoritmos de computação evolutiva e sobre o funcionamento e geração de energia em uma UHE. O terceiro capítulo propõe a metodologia adotada para a realização do trabalho. O quarto capítulo apresenta o algoritmo multiobjetivo híbrido desenvolvido neste trabalho. O quinto capítulo apresenta a proposta de modelo multiobjetivo de geração de energia com usinas em cascata. O sexto capítulo apresenta e discute os resultados dos experimentos realizados neste trabalho. O último capítulo apresenta as conclusões do trabalho e sugere possíveis trabalhos futuros.

2 Fundamentação teórica

2.1 Otimização mono-objetivo

A otimização é um processo que visa encontrar a melhor solução para uma ou mais funções que modelam os objetivos de um sistema ou problema. Frequentemente, a função problema é acompanhada de funções de restrição, que representam as limitações da natureza do sistema no mundo real. A solução, portanto, refere-se aos valores dos parâmetros do sistema, para os quais a função que o modela atinge seu valor mínimo ou máximo, respeitando as restrições impostas. A busca pelos referidos parâmetros pode ser realizada a partir da solução de um problema de otimização, cuja formulação é apresentada por Bertsekas (2005) conforme se segue:

Dado um conjunto S , uma função $f : S \rightarrow \mathbb{R}$ e um subconjunto $X \subset S$, o problema de otimização:

$$\text{Minimizar } f(\mathbf{x})$$

$$\text{Sujeito a } \mathbf{x} \in \mathbf{X},$$

consiste em encontrar um elemento $x^* \in X$ (chamado de elemento minimizador ou uma solução ótima) tal que:

$$f(x^*) \leq f(x) \text{ para todo } \mathbf{x} \in \mathbf{X}.$$

Pode-se escrever o elemento x^* , da forma:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbf{X}}{\operatorname{argmin}} f(\mathbf{x}).$$

Quando há restrições o problema torna-se:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbf{X}}{\operatorname{argmin}} f(\mathbf{x}),$$

$$\text{Sujeito a } h_1 = 0, \dots, h_m = 0, g_1 \leq 0, \dots, g_r \leq 0,$$

$$\mathbf{x}_i^{\inf} \leq \mathbf{x}_i \leq \mathbf{x}_i^{\sup}$$

em que h_i e g_j são funções de restrição de igualdade e desigualdade respectivamente do tipo $\mathbb{R}^n \rightarrow \mathbb{R}$. E o intervalo $x_i^{\inf} \leq x_i \leq x_i^{\sup}$, impõe limites superior e inferior sobre x , criando um espaço para as variáveis de decisão x_i .

É possível representar um problema de maximização como um problema de minimização da negativa da função original a ser maximizada. Por padrão otimização é definida como minimização conforme a notação aqui adotada.

Com esta formulação, $f(x)$ seria uma função que modela matematicamente o sistema para atingir o objetivo desejado, sendo chamada, portanto, de função objetivo. As funções h_i e g_j , por sua vez, modelam as limitações do sistema, enquanto o elemento x^* corresponde à solução ótima, contendo os parâmetros para atingir o objetivo desejado.

2.1.1 Otimização multiobjetivo

Na formulação anterior, a otimização é feita sobre uma única função objetivo. Nestes casos, dizemos que se trata de uma otimização mono-objetivo. Quando múltiplas funções objetivo precisam ser otimizadas simultaneamente, temos uma otimização multi-objetivo (MO). Com base na formulação da otimização mono-objetivo e de acordo com as ideias de Deb (2001), podemos definir um problema MO da seguinte maneira:

$$\begin{aligned} \mathbf{x}^* &= \underset{\mathbf{x} \in \mathbf{X}}{\operatorname{argmin}} (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})), \\ \text{Sujeito a } h_1 &= 0, \dots, h_m = 0, g_1 \leq 0, \dots, g_r \leq 0, \\ \mathbf{x}_i^{\inf} &\leq \mathbf{x}_i \leq \mathbf{x}_i^{\sup}. \end{aligned}$$

em que cada $f_i(x)$ é uma função que modela um objetivo do problema e n é o número total de objetivos.

Desse modo, um problema multiobjetivo envolve o mapeamento de variáveis de um espaço de decisão para o espaço objetivo criado pelas funções-objetivo.

2.1.1.1 Dominância entre soluções e pareto-ótimo

Por tratar múltiplas funções-objetivo simultaneamente, uma solução possui valores para cada uma das N funções-objetivo. Para determinar se uma solução é melhor que outra, é necessário considerar o conceito de dominância, que Deb (2001) define pelas regras a seguir:

Sejam A e B duas soluções para um problema MO com N funções-objetivo, então A domina B se as duas condições seguintes são verdadeiras:

1. A solução A não é pior que a solução B para todos os objetivos, ou seja $f_j(A) \leq f_j(B)$ para todo $j = 1, 2, \dots, N$;
2. A solução A é melhor que a solução B em pelo menos um objetivo, ou seja $f_k(A) < f_k(B)$ para pelo menos um $k \in \{1, 2, \dots, N\}$.

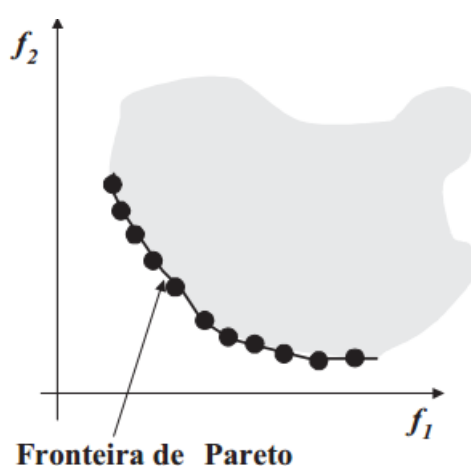
Se qualquer uma das duas condições é violada, a solução A não domina B . Quando A domina B , podemos usar a seguinte notação matemática: $A \preceq B$.

As funções-objetivo de um problema MO são em geral conflitantes entre si, pois a melhora do valor de uma função pode causar uma piora do valor de outra função. Um

exemplo prático sugerido por Deb (2001) seria a otimização da compra de um carro, em que as funções objetivos são minimizar o valor da compra e maximizar o conforto do carro. No referido exemplo, um carro mais barato implica também em maior desconforto, ou seja, um carro que melhora a função do valor de compra, piora a função do conforto.

Devido a essa relação conflitante entre as funções-objetivo e o conceito de dominância entre soluções, um problema MO possui um conjunto de soluções pareto ótimas, ao invés de uma única solução, como seria em um problema mono-objetivo. Tal conjunto de soluções, portanto, forma no espaço uma fronteira, como explicitado na Figura 2. Essa fronteira recebe o nome de Pareto-ótimo.

Figura 2 – Fronteira Pareto-ótimo em um espaço objetivo bidimensional.



Fonte: Deb (2001).

2.1.2 Otimização dinâmica

Na otimização dinâmica, o tempo é um fator considerado nas instâncias do problema, nas funções objetivo ou nas funções de restrição, consequentemente o modelo e sua solução ótima se alteram de acordo com a variação do tempo. O objetivo da otimização dinâmica não é apenas encontrar uma solução ótima, mas também acompanhá-la ao longo do tempo da forma mais próxima possível. Assim como na otimização estática, a otimização dinâmica pode ser mono-objetivo ou multiobjetivo.

Segundo Carter (2001), três abordagens são aplicadas para solucionar problemas de otimização dinâmica: a baseada em cálculo variacional, que é a técnica mais aplicada para tratar modelos com tempo contínuo; A programação dinâmica, desenvolvida principalmente para otimização com o tempo discretizado; E a abordagem baseada na extensão das técnicas de Lagrangeano na otimização estática para a otimização dinâmica.

2.2 Computação evolutiva

A computação evolutiva é uma área da inteligência computacional que se inspira em mecanismos da adaptação e evolução dos seres vivos na natureza, para criar meta-heurísticas que mantem um conjunto de soluções, em que cada solução imita o comportamento evolutivo de um indivíduo em uma população (HENGGELE; RICARDO; ANTUNES, 2013; FRANK; STEPONAVICE; REBENNACK, 2012).

Nas últimas duas décadas diversos algoritmos da computação evolutiva e outros métodos não determinísticos de otimização têm sido aplicado aos problemas de OPF e energia. Métodos evolutivos possuem maior capacidade de busca global dos que métodos convencionais determinísticos, além disso estes algoritmos são bem adequados para problemas MO em que se deseja um conjunto de soluções em vez de apenas uma única solução (FRANK; STEPONAVICE; REBENNACK, 2012).

Neste trabalho são abordados os algoritmos mono-objetivo: Evolução Diferencial (DE, do termo original do inglês *Differential Evolution*) de Storn e Price (1997), Otimização por Enxame de Partículas (PSO, do termo original do inglês *Particle Swarm Optimization*) de Kennedy e Eberhart (1995), Otimização por Enxame de Partículas Evolucionário (EPSO, do termo original do inglês *Evolutionary Particle Swarm Optimization*) de Miranda e Fonseca (2002) e o *Differential Evolutionary Particle Swarm Optimization* (DEEPSO) de Miranda e Alves (2013). Estes algoritmos foram a inspiração para a criação do C-DEEPSO e hC-DEEPSO de Marcelino (2017).

Além dos algoritmos já mencionados também são abordados os algoritmos MO: *Nondominated Sorting Genetic Algorithm II* (NSGA-II) de Deb (2002), *Strength Pareto Evolutionary Algorithm 2* (SPEA2) de Zitzler, Laumanns e Thiele (2001), Otimização por Enxame de Partículas Multiobjetivo (MOPSO, do termo original do inglês *Multi-objective Particle Swarm Optimization*) de Coello e Lechuga (2002) e Evolução Diferencial Multiobjetivo (MODE, do termo original do inglês *Multi-objective Differential Evolution*) de Xue, Sanderson e Graves (2003) e o *Multi-objective evolutionary particle swarm optimization* (MEPSO) de (MACIEL et al., 2012); tais algoritmos foram usados como comparação e também alguns de seus operadores serviram de inspiração para o algoritmo híbrido MO proposto neste trabalho.

Neste trabalho, um dos modelos de geração de energia com UHE a ser otimizado é de natureza dinâmica. Embora exista na literatura estratégias e técnicas para computação evolutiva em otimização dinâmica (NGUYEN; YANG; BRANKE, 2012). Estas não serão aplicadas em nenhum dos algoritmos avaliados. A otimização dinâmica será feita com a estratégia de reinicialização, na qual sempre que há uma mudança temporal no modelo do problema, o algoritmo começa uma nova otimização estática sem considerar soluções de otimizações em tempos anteriores. Justifica-se o uso dessa estratégia pela simplicidade e

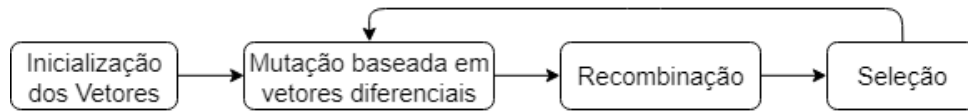
por não haver necessidade alterar o funcionamento dos algoritmos.

2.2.1 Algoritmos mono-objetivo

2.2.1.1 Evolução diferencial (DE)

Proposto por [Storn e Price \(1997\)](#), o DE é uma metaheurística de busca estocástica para problemas de otimização em um espaço contínuo. Apesar de não ter inspiração em nenhum mecanismo da natureza, o DE funciona de maneira semelhante a outros algoritmos de computação evolutiva, mantendo uma população em que seus indivíduos são soluções para o problema a ser otimizado. Essa população passa pelos processos de mutação, recombinação e seleção gerando à cada iteração uma nova população. Cada iteração do algoritmo pode ser chamada de uma geração da população. Este funcionamento do DE é ilustrado pela [Figura 3](#).

Figura 3 – Funcionamento do DE.



Fonte: Adaptado de [Storn e Price \(1997\)](#).

A população inicial é usualmente gerada de maneira aleatória com uma distribuição uniforme nos limites das variáveis de decisão do problema a ser otimizado. Caso exista uma solução preliminar a população inicial pode ser gerada adicionando desvios aleatórios sobre esta solução.

Uma vez inicializada a população o algoritmo gera novos indivíduos por meio do operador de mutação diferencial. Este operador calcula a diferença entre os vetores de dois indivíduos distintos e em seguida adiciona esta diferença ao vetor de um terceiro indivíduo, conforme definido pela [Equação \(1\)](#),

$$\mathbf{V} = \mathbf{X}_{r1} + \mathbf{F} * (\mathbf{X}_{r2} - \mathbf{X}_{r3}), \quad (1)$$

na qual X_{r1}, X_{r2}, X_{r3} são os vetores de indivíduos da população atual e V é um novo indivíduo denominado de mutante. Os índices $r1, r2, r3$ são inteiros aleatórios e mutualmente diferentes no intervalo de $[1 - NP]$ em que NP é o número total de indivíduos na população. Este operador de mutação é usado para cada um dos indivíduos da população e portanto os índices $r1, r2, r3$ também são mutualmente diferentes do índice do indivíduo atual no operador. F é um valor real geralmente no intervalo $[0 - 1]$ que modifica a intensidade do fator diferencial $(X_{r2} - X_{r3})$.

Com o propósito de aumentar a diversidade das soluções aplica-se o operador de recombinação sobre a população atual e a população de mutantes. Este operador cria uma nova população em que seus indivíduos são chamados de indivíduos teste. Cada indivíduo teste é formado de acordo com a [Equação \(2\)](#),

$$U = \begin{cases} V, & \text{se } U_{[0-1]} \leq C_r \vee j = \delta \\ X, & \text{caso contrário} \end{cases}, \quad (2)$$

na qual U é o vetor do indivíduo teste, V é o vetor do indivíduo mutante e X é o vetor do indivíduo da população atual, j indica o índice do elemento nos vetores. C_r é a constante de recombinação definida pelo usuário no intervalo $[0 - 1]$, valores mais próximo de 1 possuem maior probabilidade de que elementos do vetor mutante serão herdados e para C_r igual a 1 implica que o indivíduo de teste é igual mutante. $U_{[0,1]}$ é um número aleatório escolhido dentro da distribuição uniforme no intervalo de 0 a 1. δ é um índice aleatório que garante que pelo menos um dos elementos do vetor de teste será herdado do vetor mutante.

O indivíduo teste é então comparado com seu respectivo indivíduo da população atual pelo operador de seleção para definir qual indivíduo permanecerá para a próxima geração, neste caso comumente seleciona-se o indivíduo com melhor avaliação pela função de aptidão que quantifica a qualidade da solução de um indivíduo para o problema de otimização a ser resolvido. Após a seleção o algoritmo finaliza, caso um critério de parada seja satisfeito, como por exemplo o número de iterações máximo ou que o número de iterações sem melhoria seja atingido, caso contrário o algoritmo retoma o passo de mutação diferencial.

A descrição anterior do algoritmo DE é a proposta inicial feita por [Storn e Price \(1997\)](#). Devido ao seu bom desempenho o DE teve boa aceitação por pesquisadores e com o tempo diferentes estratégias evolutivas foram elaboradas criando a notação $DE/X/Y/Z$ sendo:

- X especifica o vetor a ser mutado, por exemplo para X como “rand” um vetor aleatório é escolhido, para X como “best” escolhe-se o vetor do indivíduo com melhor avaliação pela função objetivo;
- Y define o número de vetores para a fator diferencial no operador de mutação;
- Z determina o tipo de cruzamento, por exemplo o cruzamento definido anteriormente é o cruzamento do tipo “bin” de binomial.

Utilizando esta notação a forma clássica do DE seria DE/Rand/1/Bin, a exemplo de outras estratégias temos os trabalhos [Price \(1999\)](#), [Price, Storn e Lampinen \(2006\)](#) com as estratégias: DE/best/1, DE/target-to-best/1, DE/best/2, DE/rand/2 e DE/current-to-rand/1.

2.2.1.2 Otimização por enxame de partículas (PSO)

Assim como o DE, o PSO proposto por [Kennedy e Eberhart \(1995\)](#) é uma meta-heurística de busca estocástica com populações para funções contínuas e não-lineares. Sua inspiração vem do comportamento social de animais quando estão se movimentam em conjuntos tais como bandos, rebanhos, revoadas, exames e outros. No PSO os indivíduos da população são denominados de partículas e cada partícula se desloca no espaço pela a seguinte regra:

$$\mathbf{X}_i^{\text{novo}} = \mathbf{X}_i + \mathbf{V}_i, \quad (3)$$

na qual $\mathbf{X}_i$ é o vetor dos parâmetros da partícula e $\mathbf{V}_i$ é sua velocidade que é definida pela [Equação \(4\)](#) :

$$\mathbf{V}_i^{\text{novo}} = D(t)\mathbf{W}_{i0} * \mathbf{V}_i + U_1 * \mathbf{W}_{i1}(\mathbf{B}_i - \mathbf{X}_i) + U_2 * \mathbf{W}_{i2}(\mathbf{B}_g - \mathbf{X}_i), \quad (4)$$

$D(t)$ é uma função que decresce à medida que o número de iterações do PSO aumenta, W_{i0}, W_{i1}, W_{i2} são pesos definidos pelos usuário. B_i é o vetor da melhor solução encontrada pela partícula, B_g é o vetor da melhor partícula encontrada pela população ou seja a melhor solução encontrada pelo PSO até o momento. U_i são números aleatórios amostrados de uma distribuição normal no intervalo de 0 a 1. Dado esta equação o PSO é descrito pelo [Algoritmo 1](#).

Algoritmo 1: Pseudo-código do funcionamento do PSO

```

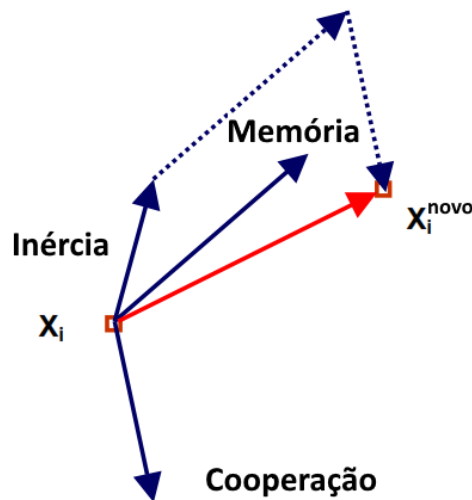
Inicialize as partículas  $X(i)$  aleatoriamente dentro dos limites do espaço de
busca.
Atualize a melhor solução encontrada pela partícula,  $B_i = X_i$ .
if  $f(X_i) < f(B_g)$  then
    |  $B_g = X_i$ 
end
Inicialize as velocidades das partículas  $V_i$  aleatoriamente.
while Critério de parada não satisfeito do
    | Atualize a velocidade  $V_i$  das partículas de acordo com a Equação \(4\)
    | Atualize a posição  $X_i$  das partículas de acordo com a Equação \(3\)
    | if  $f(X_i) < f(B_i)$  then
    | |  $B_i = X_i$ 
    | end
    | if  $f(X_i) < f(B_g)$  then
    | |  $B_g = X_i$ 
    | end
end

```

Por meio da [Equação \(4\)](#) de velocidade das partículas podemos interpretar as três parcelas da soma da seguinte maneira: a primeira parcela é um termo de inercia já

que se baseia em sua velocidade anterior, a segunda parcela é um termo de memória da partícula por considerar a melhor solução encontrado pela partícula e a terceira parcela é o termo de cooperação por considerar a melhor solução encontrada pela população. A [Figura 4](#) ilustra o comportamento da velocidade e o efeito dos três termos.

Figura 4 – Contribuição dos termos na equação de movimento do PSO.



Fonte: Adaptado de [Miranda e Fonseca \(2002\)](#).

2.2.1.3 Otimização por enxame de partículas evolucionário (EPSO)

O EPSO é uma meta-heurística que combina características da estratégia evolucionária com o PSO. Criado por [Miranda e Fonseca \(2002\)](#) o EPSO visa tornar o PSO um algoritmo mais auto adaptativo e, conseqüentemente, retirar a responsabilidade do usuário em definir parâmetros do algoritmo que muitas vezes são sensíveis ao problema a ser resolvido. EPSO adiciona técnicas evolucionárias sobre os parâmetros do PSO para torna-los auto-adaptativo e um novo procedimento para seleção de partículas.

Cada iteração do EPSO segue as seguintes etapas:

1. Replicação: cada partícula é clonada R vezes.
2. Mutação: cada partícula tem seus pesos W mutados.
3. Reprodução: as partículas mutadas geram seus descendentes seguindo uma regra de movimento.
4. Avaliação: os descendentes são avaliados pela função de aptidão.
5. Seleção: por meio do torneio estocástico as melhores partículas são selecionadas para a próxima geração.

A regra de movimento do EPSO segue o mesmo padrão da regra do PSO, dado uma partícula X_i , sua próxima posição X_i^{novo} é a soma de X_i com a sua nova velocidade X_i^{novo} , porem o cálculo da nova velocidade no EPSO segue a fórmula da [Equação \(5\)](#),

$$V_{i \text{ novo}} = W_{i0}^* \cdot V_i + W_{i1}^* (B_i - X_i) + W_{i2}^* (B_g^* - X_i), \quad (5)$$

na qual B_i e B_g são equivalentes às suas respectivas variáveis na [Equação \(4\)](#) do PSO, $W_{i,j}$ são os pesos. No EPSO a função decrescente para a velocidade atual é removida. O símbolo $*$ indica que aquela variável sofre uma mutação, no caso dos pesos W_{ij} a mutação é feita de acordo com a [Equação \(6\)](#), já para a mutação do melhor global B_g é feita pela [Equação \(7\)](#).

$$W_{i,k}^* = W_{i,k} + \tau N(0,1), \quad (6)$$

$$B_g^* = B_G + \tau' N(0,1). \quad (7)$$

Na [Equação \(6\)](#) e [Equação \(7\)](#), $N(0,1)$ é um valor aleatório amostrado da distribuição Gaussiana com média 0 e variância 1, τ e τ' são dois parâmetros de aprendizado que podem ser fixos ou tratados como variáveis sujeitas à mutação.

2.2.1.4 Differential evolutionary particle swarm optimization (DEEPSO)

O EPSO foi aprimorado por [Miranda e Alves \(2013\)](#) por meio da adição de partes do operador de mutação do DE à equação de movimento do EPSO, criando assim uma nova meta-heurística híbrida denominada DEEPSO. O algoritmo do DEEPSO segue os mesmos passos do algoritmo EPSO porem a fórmula de atualização da velocidade segue a [Equação \(8\)](#).

$$V_{i \text{ novo}} = W_{i0}^* V_i + W_{i1}^* (X_{r1} - X_i) + P[W_{i2}^* (B_g^* - X_i)] \quad (8)$$

Nessa nova equação parte do termo de memória do EPSO é substituído pelo vetor de diferenças do DE onde X_{r1} é uma partícula distinta da atual X_i , este novo termo passa a se chamar termo de memória. O termo de cooperação também é alterado, nele a variável P é uma matriz diagonal de tamanho igual à dimensão dos vetores das partículas. Os elementos da diagonal são valores binários definidos de acordo com uma probabilidade. Caso o valor seja 1, tem probabilidade p e o valor 0 com probabilidade $(1 - p)$. Com issom partes do termo de cooperação podem contribuir para a velocidade, nos casos extremos em que P é a matriz identidade temos uma contribuição completa ou caso P seja uma matriz

nula não há o termo de cooperação. A mutação da partícula com melhor global B_g^* segue a [Equação \(9\)](#) onde W_g é um peso mutável.

$$B_g^* = B_G(1 + W_g N(0,1)). \quad (9)$$

No termo de memória os vetores X_{r1} e X_i devem estar ordenados de acordo com os seus valores de aptidão, se $f(X_{r1}) < f(X_i)$ mantém-se a ordem descrita pela [Equação \(8\)](#), caso contrário deve-se trocar a posição desses vetores na equação. [Miranda e Alves \(2013\)](#) propuseram quatro estratégias para escolher a partícula X_{r1} :

- Sg : amostrado a partir de todas as partículas da geração atual;
- PB : amostrado da população em memória B das melhores partículas do Passado (gerações anteriores);
- $Sg - rnd$: extraído como uma recombinação uniforme de partículas da geração atual, e;
- $PB - rnd$: extraído como uma recombinação uniforme da memória B .

2.2.1.5 Canonical differential evolutionary particle swarm optimization (C-DEEPSO)

O C-DEEPSO de ([MARCELINO et al., 2018](#)) é uma melhoria do DEEPSO que se inspira na versão canônica do DE como descrito na [Equação \(1\)](#). Em C-DEEPSO o movimento é feito de acordo com [Equação \(10\)](#) e [Equação \(11\)](#),

$$V_i^{novo} = W_I^* V_i + W_A^* (X_{st} - X_i) + W_C^* C(B_g^* - X_i), \quad (10)$$

$$X_{st} = X_r + F(X_{best} - X_r) + F(X_{r1} - X_{r2}). \quad (11)$$

O termo de assimilação, $W_A^*(X_{st} - X_i)$, do C-DEEPSO incorpora mais elementos da estrutura dos operadores de mutação e recombinação do DE. O vetor X_{st} é calculado de acordo com a [Equação \(11\)](#) que adota a estratégia evolutiva do DE *current-to-best* ([QIN; SUGANTHAN, 2005](#); [HUANG; QIN; SUGANTHAN, 2006](#)).

Na [Equação \(11\)](#) F é a taxa de perturbação no intervalo $[0 - 1]$ conforme definido na mutação do DE. X_{r1} e X_{r2} são soluções aleatoriamente amostradas. O vetor X_r é gerado utilizando a nova estratégia $S_g P_b - rnd$. Esta estratégia soma vetorialmente a solução de uma partícula aleatória da população atual com a solução de uma partícula da memória B . B possui um conjunto das melhores soluções encontradas pelo algoritmo. W_I , W_A e W_C são respectivamente os pesos de inércia, assimilação e cooperação e C é uma matriz equivalente à matriz P do DEEPSO. Assim como em EPSO e DEEPSO o símbolo * indica

que a variável sofre mutação, os pesos sofrem mutação da mesma maneira que em EPSO, já a mutação da partícula B_g segue a [Equação \(12\)](#).

$$B_g^* = B_G(1 + \tau N(0,1)). \quad (12)$$

Após o cálculo de X_{st} , X_r e X_{st} são comparados quanto aos seus respectivos valores de aptidão, caso o valor apresentado por X_{st} seja maior que X_r , então X_{st} é substituído por X_r na equação de movimento, caracterizando assim o movimento de recombinação presente no DE. O C-DEEPSO também pode utilizar outras estratégias evolucionárias do DE, como por exemplo no trabalho de [Marcelino et al. \(2016\)](#) adotou-se o $DE/best/1$ e o movimento das partículas é realizado pela [Equação \(13\)](#). O [Algoritmo 2](#) mostra o funcionamento do C-DEEPSO.

$$V_{i \text{ novo}} = W_I^* V_i + W_A^* (X_{\text{best}} + F(X_r - X_i)) + W_C^* C(B_g^* - X_i) \quad (13)$$

Algoritmo 2: Pseudo-código do funcionamento do C-DEEPSO

```

Inicialize aleatoriamente a população de  $NP$  partículas nos intervalos do espaço
de busca e o contador de gerações  $t = 1$ 
Avalie a população atual,  $P_t$ 
Atualize a melhor solução obtida até o momento  $X_g$ 
while Critério de parada não é satisfeito do
  for cada partícula  $i$  pertencente a população  $P_t$  do
    Calcular  $X_r$  usando a estratégia  $S_g P_b - rnd$ 
    Clone  $X_i$ 
    Mutar os parâmetros  $W_I, W_A, W_C$  usando a Equação \(6\)
    Mutar  $X_b^*$  usando a Equação \(12\)
    Aplicar a regra de movimento da Equação \(10\) em  $X_i$  e seu clone
    Avaliar  $X_i$  e seu clone
    Selecionar a partícula com melhor aptidão para a geração seguinte
  end
  Atualizar o melhor indivíduo  $X_g$  e a memória  $B$   $t = t + 1$ 
end

```

No trabalho de [Marcelino et al. \(2018\)](#) foram feitas alterações na etapa do cálculo da variável X_{st} que trouxeram um melhor desempenho para C-DEEPSO. Nestas versões do C-DEEPSO, o X_{st} funciona como a melhor solução encontrada pela partícula. Ao termino do cálculo de um novo X_{st} , este é comparado com a melhor solução encontrada anteriormente pela partícula, caso a aptidão do novo X_{st} seja pior, o X_{st} será uma cópia do melhor anterior, caso contrário a solução encontrada se mantém como X_{st} .

Foi adicionado um elitismo ao subconjunto de vetores para serem escolhidos como as variáveis X_r , implicando que apenas as soluções com aptidão superior à aptidão da partícula são amostradas. Como esta alteração restringe a quantidade de amostragens

viáveis, é possível que o tamanho do subconjunto seja insuficiente para o cálculo da variável, para esta situação o valor do X_{st} será o valor da melhor solução encontrada anteriormente pela partícula.

Ao termino do cálculo do X_{st} , um operador de recombinação é aplicado à variável. Neste operador cada posição do vetor X_{st} tem uma probabilidade de ser modificado, quando esta probabilidade é satisfeita, o valor da posição é substituído pelo valor do melhor global ou do melhor local na mesma posição.

2.2.2 Algoritmos multiobjetivo

2.2.2.1 *Nondominated sorting genetic algorithm II* (NSGA-II)

Proposto por Deb (2002), o NSGA-II é um algoritmo genético (AG) capaz de solucionar problemas MO por meio de uma ordenação elitista por dominância. O AG é inspirado nos mecanismos de seleção natural e genética natural. Assim como o DE o AG opera sobre uma população de indivíduos que são vetores que representam soluções para o problema a ser otimizado.

Após a criação de uma população inicial, o funcionamento do AG segue quatro etapas: a avaliação da aptidão da população atual; a reprodução dos indivíduos mais aptos; a recombinação com os indivíduos definidos na etapa de reprodução e a mutação sobre os indivíduos gerados na recombinação. Ao fim de cada iteração dessas quatro etapas, temos uma nova geração da população, o algoritmo permanece criando novas gerações até um critério de parada ser satisfeito. A Figura 5 apresenta um fluxograma de um AG típico.

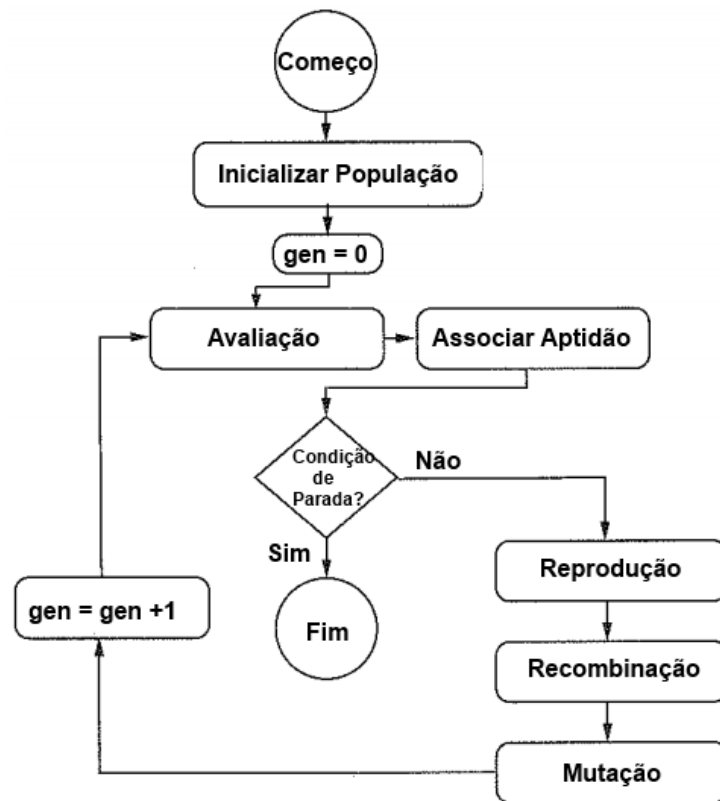
O funcionamento de cada etapa do AG é feito por um operador. Alguns operadores permitem diferentes maneiras de implementação, porem o seu objetivo principal não é alterado para cada implementação.

O operador de associação de aptidão tem como objetivo avaliar o quão adequada uma solução é para o problema a ser otimizado. O valor da aptidão do indivíduo é calculado pela função de aptidão, que é baseada nas funções objetivo e de restrição do modelo do problema, de modo que as melhores soluções tenham melhores valores.

O operador de reprodução, também chamado de operador de seleção, escolhe um conjunto de N indivíduos que possuam bons valores de aptidão, os quais serão usados pelo operador de recombinação. Em geral, os indivíduos escolhidos são chamados de pais e os indivíduos gerados pelo operador de recombinação são chamados de filhos. Um exemplo de operador de reprodução é o torneio binário, onde duas soluções são selecionadas aleatoriamente e aquela que possuir melhor aptidão é escolhida como pai.

A partir dos indivíduos escolhidos pelo operador de reprodução, o operador de recombinação (ou de cruzamento) cria novos indivíduos com base nas características

Figura 5 – Etapas de funcionamento de um algoritmo genético típico.



Fonte: Adaptado de Deb (2001).

daqueles escolhidos. Este operador é o principal responsável pela capacidade de busca dos AG devido à criação de novos pontos de busca. Como exemplo temos o operador de cruzamento convexo por coordenada que utiliza dois pais para gerar dois filhos pela seguinte formula:

$$Filho1_i = Pai1_i * \alpha_i + (1 - \alpha_i) * Pai2_i,$$

$$Filho2_i = Pai2_i * \alpha_i + (1 - \alpha_i) * Pai1_i,$$

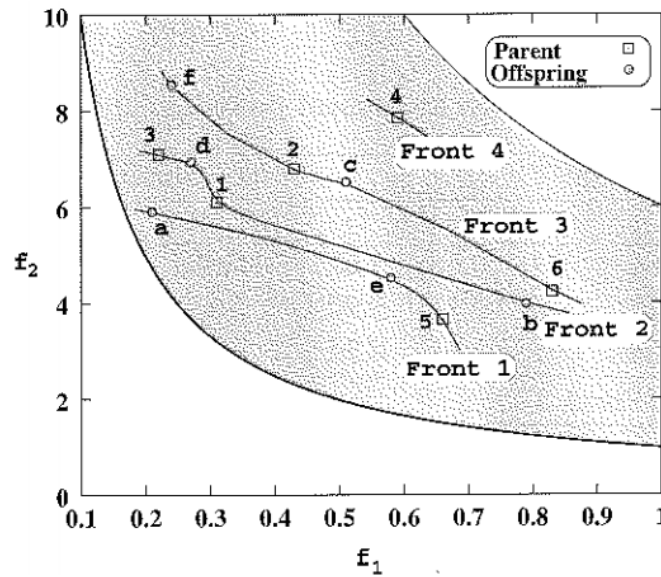
em que i é o i -ésimo elemento de um vetor e α é um vetor de valores reais gerados de forma aleatória no intervalo entre 0 e 1.

Para aumentar a diversidade na população e expandir a área de busca para outras regiões aplica-se o operador de mutação. Para tal, o operador altera arbitrariamente valor dos indivíduos filhos, em uma mutação uniforme um dos elementos do vetor do filho é escolhido aleatoriamente e seu valor é substituído por outro número aleatório dentro de um intervalo definido.

Além dos operadores tradicionais, o NSGA-II acrescenta dois operadores para lidar com problemas MO: a Ordenação por Dominância Rápida e a Atribuição de Distância

de Multidão. O primeiro operador separa as soluções de uma população P em fronteiras $F1, F2, \dots, Fi$, de tal forma que a fronteira $F1$ contenha as soluções não dominadas do conjunto P , a fronteira $F2$ contenha as soluções dominadas pelo conjunto $P - F1$, a fronteira $F3$ contenha as soluções não dominadas no conjunto $P - (F1 \cup F2)$ e assim sucessivamente. A Figura 6 exemplifica um caso com quatro fronteiras identificadas no espaço objetivo.

Figura 6 – Fronteiras de soluções no espaço objetivo.



Fonte: Deb (2001).

Para seu funcionamento, o NSGA-II atribui três novas variáveis para cada solução p da população: um contador n_p contendo o número de soluções que dominam p ; um conjunto S_p contendo as soluções que são dominadas por p ; e p_{rank} que identifica a qual fronteira a solução pertence. O cálculo das três variáveis e a criação da primeira fronteira são feitos pelo Algoritmo 3 adaptado de Deb (2002). Uma vez calculado os valores de S_p , n_p e a primeira fronteira $F1$, podemos organizar as demais soluções nas outras fronteiras Fi , utilizando o Algoritmo 4 adaptado de Deb (2002).

O operador de atribuição de distância de multidão é utilizado no NSGA-II para manter a diversidade da população. Este operador emprega uma estimativa da densidade das soluções que rodeiam cada solução da população. Para isso, calcula-se o valor de distância de multidão para cada solução pela média das distâncias das duas soluções adjacentes para todos os objetivos do problema. O Algoritmo 5 adaptado de Deb (2002) mostra os passos para o cálculo da distância de multidão de cada solução de uma fronteira não dominada.

A distância de multidão permite que soluções mais dispersas possam ser selecionadas durante o operador de seleção. Com essa abordagem a solução vencedora do

Algoritmo 3: Pseudo-código para o cálculo das variáveis n_p , S_p , p_{rank} e da criação da primeira fronteira

```

for cada solução  $p$  da população  $P$  do
     $S_p = \emptyset$ 
     $n_p = 0$ 
    for cada solução  $q$  da população  $P$  do
        if  $p$  domina  $q$  then
             $S_p = S_p \cup q$  // Se  $p$  domina  $q$ , adicionar  $q$  para o conjunto de
            soluções que  $p$  domina
        end
        else
             $n_p = n_p + 1$  // Caso contrário incrementar o contador de soluções
            que dominam  $p$ 
        end
    end
    if  $n_p = 0$  then
        // se nenhuma outra solução domina  $p$ , então  $p$  pertence a primeira
        fronteira  $F_1$ 
         $p_{rank} = 1$ 
         $F_1 = F_1 \cup p$ 
    end
end

```

Algoritmo 4: Pseudo-código para criação das demais fronteiras

```

 $i = 1$  // Inicializa o contador de fronteira
while  $F_i \neq \emptyset$  do
     $Q = \emptyset$  // Conjunto para armazenar membros da próxima fronteira
    for cada  $p \in F_i$  do
        for cada  $q \in S_p$  do
             $n_q = n_q - 1$ 
            if  $n_q = 0$  then
                 $q_{rank} = i + 1$  //  $q$  pertence à próxima fronteira
                 $Q = Q \cup q$ 
            end
        end
    end
     $i = i + 1$ 
     $F_i = Q$ 
end

```

torneio é aquela que está na fronteira menos dominada, e no caso que ambas soluções pertencem à mesma fronteira, a vencedora é aquela que tiver maior distância de multidão.

O funcionamento do NSGA-II é bastante similar ao de um AG, o primeiro passo é a geração de uma população inicial P_0 aleatória de tamanho N . Em seguida, as soluções de P_0 são classificadas quanto às fronteiras não dominadas às quais pertencem. Com P_0 classificada, gera-se a população de filhos Q_0 de tamanho N pela seleção por torneio,

Algoritmo 5: Pseudo-código para o cálculo da distância de multidão

```

 $L$  = tamanho da fronteira  $F$ 
for cada solução  $p \in F$  do
  |  $p_{\text{distância}} = 0$ 
end
for cada objetivo  $M$  do
  |  $I = \text{ordenação}(F, M)$  // Ordene as soluções pelo seu valor no objetivo atual
  |  $I[1] = I[L] = \infty$  // Desta forma as soluções nos extremos da fronteira são
  | sempre selecionadas
  | for  $k = 2$  até  $L - 1$  do
  | |  $p_{\text{distância}} = p_{\text{distância}} + (F[k + 1]_M - F[k - 1]_M) / (F_M^{\text{Max}} - F_M^{\text{Min}})$ 
  | end
end

```

cruzamento e mutação sobre P_0 .

Após a criação das populações P_0 e Q_0 , de acordo com Deb (2001), o funcionamento de cada iteração do NSGA-II pode ser descrito pelos seguintes passos e ilustrado pela Figura 7:

1. Combine a população P_t e Q_t em R_t , utilize a ordenação por dominância rápida sobre P_t e gere o conjunto das fronteiras não dominadas F .
2. Inicialize a população P_{t+1} como vazia e o contador i como 0.
 - 2.1. Enquanto o tamanho de $P_{t+1} \cup F_i \leq N$ faça $P_{t+1} = P_{t+1} \cup F_i$ e $i = i + 1$.
3. Realize a atribuição de distância de multidão sobre F_i . Ordene as soluções pela distância de multidão e adicione as $(N - |p_{t+1}|)$ soluções com maior distância de multidão em P_{t+1} .
4. Gere a população de filhos Q_{t+1} pela seleção por torneio com distância de multidão, cruzamento e mutação sobre P_{t+1} .

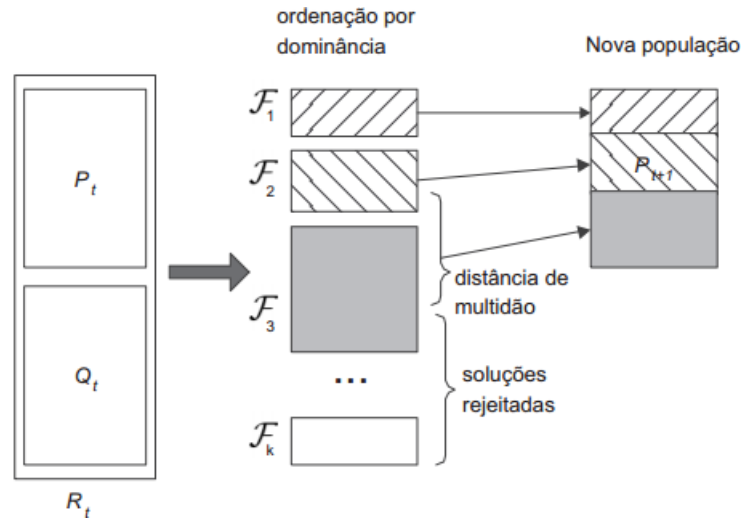
As iterações se repetem até que um critério de parada seja satisfeito.

2.2.2.2 Strength pareto evolutionary algorithm 2 (SPEA2)

O SPEA2 de Zitzler, Laumanns e Thiele (2001) assim como o NSGA-II é um AE que soluciona problemas MO por meio de técnicas que tratam de dominância e pareto-ótimo. Além de uma população, o SPEA2 mantém um conjunto externo denominado de arquivo, que persiste uma cópia dos indivíduos que compõem a fronteira não-dominada. As gerações da população do SPEA2 são feitas de acordo com as etapas a seguir:

1. Inicialização: gere uma população inicial aleatória P_0 e crie o arquivo $P_0 = \emptyset$. defina $t = 0$
2. Associação de aptidão: calcule os valores de aptidão dos indivíduos em P_t e em $\bar{P}_t$. Detalhes do funcionamento dessa etapa serão discutidos nos parágrafos seguintes.
3. Seleção Ambiental: copie todos os indivíduos não dominados nas populações P_t e $\bar{P}_t$

Figura 7 – Principais etapas de uma iteração do NSGA-II.



Fonte: Deb (2001).

para $\bar{P}_{t+1}$. Caso o tamanho de $\bar{P}_{t+1}$ seja maior que $\bar{N}$, então reduza $\bar{P}_{t+1}$ utilizando o operador de truncamento. Detalhes dessa etapa serão discutidos nos parágrafos seguintes.

4. Finalização: caso $t \geq T$ ou outro critério de parada seja satisfeito, então o conjunto de saída A será os indivíduos no arquivo $\bar{P}_{t+1}$ e o algoritmo finaliza.
5. Seleção para Reprodução: aplique o torneio binário nos indivíduos do arquivo $\bar{P}_{t+1}$ para selecionar os pais que sofreram operações evolutivas como cruzamento e mutação para gerar a população P_{t+1} . Retorne o algoritmo ao passo 2.

No qual N é a tamanho da população $\bar{N}$ é o tamanho do arquivo, T é o número máximo de gerações e A é o conjunto de soluções não dominadas e a saída do algoritmo, t é o índice da geração.

A associação de aptidão do SPEA2 para cada indivíduo i é baseada no número de indivíduos que dominam e são dominados por i . Cada indivíduo i no arquivo $\bar{P}_t$ e na população P_t terá a ele atribuído um valor $S(i)$, que é o número de soluções que i domina. Com os valores de $S(i)$ calcula-se a função $R(i)$, que é um somatório de $S(j)$ em que j é um indivíduo que domina i , nesta função estamos interessados nos indivíduos que a minimizam, os indivíduos não dominados são aqueles que possuem $R(i) = 0$.

Assim como o NSGA-II, o SPEA2 incorpora mecanismos para diferenciar soluções que se encontram em uma mesma fronteira pareto. O SPEA2 adota uma estimacão de densidade baseado no método de k-vizinhos mais próximos de Silverman (1986). Para cada indivíduo i a distância (no espaço objetivo) para todos os outros indivíduos j no arquivo e população, é calculada e armazenada em uma lista. Após ordenar a lista em ordem crescente, o k-ésimo elemento fornece a distância procurada para i , esse valor é denotado

de σ_i^k . No SPEA2 o valor de k adotado é $k = \sqrt{(N + \bar{N})}$ e a densidade $D_{(i)}$ de cada indivíduo é calculada por:

$$D_i = \frac{1}{\sigma_i^k + 2}. \quad (14)$$

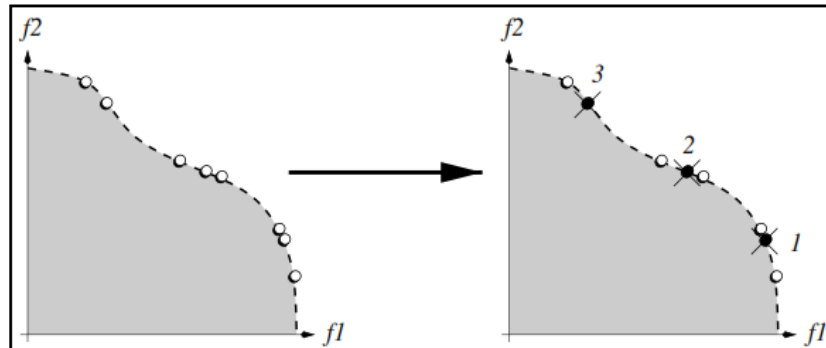
com $D_{(i)}$ e $R_{(i)}$ define-se que a aptidão de um indivíduo:

$$F_{(i)} = R_{(i)} + D_{(i)}. \quad (15)$$

A seleção ambiental é uma operação com o objetivo de se atualizar o arquivo. Todos os indivíduos que não são dominados, i.e. possuem $F(i) < 1$, devem ser copiados para o arquivo, em uma situação ideal temos que o número de indivíduos não dominados é igual ao tamanho do arquivo e a operação está completa. Caso o número de indivíduos seja menor que o tamanho $\bar{N}$ do arquivo, o espaço restante do arquivo é preenchido com os melhores indivíduos da população e do arquivo atual. Para isto estes conjuntos são ordenados pela aptidão de seus indivíduos e os $\bar{N} - |\bar{P}_{t+1}|$ primeiros indivíduos são selecionados.

Quando o tamanho do arquivo é insuficiente aplica-se um procedimento de truncamento sobre o conjunto de indivíduos não dominados. Este procedimento é feito iterativamente e em cada iteração um indivíduo é removido até que o tamanho do conjunto seja igual a $\bar{N}$. Os indivíduos removidos são aqueles que de acordo com sua distância para o seu vizinho mais próximo é mínima, ou seja, o truncamento visa manter uma boa distribuição nos indivíduos do arquivo. A Figura 8 ilustra o funcionamento do truncamento.

Figura 8 – Operador de truncamento do SPEA2, a esquerda pareto original e a direita o pareto destacando os indivíduos a serem removidos para um truncamento para $\bar{N} = 5$.



Fonte: Zitzler, Laumanns e Thiele (2001).

2.2.2.3 Abordagens multiobjetivo para otimização por enxame de partículas

O MOPSO de Coello e Lechuga (2002) é uma das primeiras adaptações do PSO para lidar com problemas MO. Utilizando um repositório para armazenar as posições das partículas não dominadas e com a geração de hipercubos que dividem o espaço de busca para guiar as partículas. O MOPSO de Coello e Lechuga (2002) funciona seguindo os seguintes passos:

1. Inicialize a população P e as respectivas velocidades $V[i] = 0$.
2. Avalie cada partícula de P .
3. Armazene as posições das partículas não dominadas no repositório R .
4. Gere os hipercubos do espaço de busca explorado até o momento e localize as partículas utilizando esses hipercubos como um sistema de coordenadas, onde cada coordenada de cada partícula é definido de acordo com seus valores nas funções objetivo.
5. Inicialize a memória de cada partícula, $P_{best}[i] = P[i]$ (esta memória é armazenada no repositório R e serve como guia para a movimentação no espaço de busca)
6. Enquanto o critério de parada não for satisfeito faça
 - 6.1. Calcule a velocidade de cada partícula pela expressão:

$$V[i] = W.V[i] + R_1(P_{best}[i] - P[i]) + R_2(R[h] - P[i]).$$

W é o peso de inércia no intervalo $[0 - 4]$. R_1 e R_2 são número aleatórios no intervalo $[0 - 1]$.

O índice h é selecionado da seguinte maneira: cada hipercubo recebe um valor de aptidão, que é calculado dividindo-se um número $x > 1$ pela quantidade de partículas que o hipercubo contém. Desta forma os hipercubos com mais partículas em seu interior recebem um valor de aptidão menor. Com estes valores de aptidão realiza-se uma seleção por roleta para se escolher um hipercubo.

A partir do hipercubo escolhido, uma de suas partículas será escolhida aleatoriamente, a partícula escolhida é o $R[h]$

- 6.2. Calcule a nova posição das partículas pela equação:

$$P[i] = P[i] + V[i]$$

- 6.3. Avalie as partículas em P .
- 6.4. Atualize o repositório R . Essa atualização consiste em colocar todas as localizações das partículas não dominadas atuais no repositório. Caso uma localização anterior seja dominada no cenário atual, esta localização é eliminada do repositório. Sempre que o repositório se encontrar cheio, as partículas que se encontram em regiões menos populosas recebem uma maior prioridade.
- 6.5. Se a posição atual da partícula for melhor que a posição mantida na memória, a memória é atualizada ($P_{best}[i] = P_i$).
- 6.6. Incremente o contador do laço de repetição.

Mostaghim e Teich (2003) discutem estratégias para se definir o melhor global para o movimento das partículas em problemas MO. Considerando que a partícula com o melhor global é guia para as demais, é essencial encontrar uma boa estratégia pois em problemas MO comumente não há apenas um melhor global.

Mostaghim e Teich (2003) propõem o método sigma que associa à cada partícula um valor σ , que é utilizado para se estimar a distância entre partículas no espaço das funções objetivo. O cálculo do σ é feito de maneira que todos os pontos que passam por uma mesma reta da origem do espaço, possuem o mesmo valor de σ .

O valor de sigma é calculado pelas combinações das diferenças do quadrado das coordenadas dividido pelo somatório das coordenadas ao quadrado, a Equação (16) exemplifica o cálculo do σ em um espaço de três dimensões. A ideia do método sigma para o MOPSO é utilizar os valores de aptidão das partículas em cada função objetivo como as coordenadas da Equação (16), desta forma cada partícula procura no repositório das partículas não dominadas, aquela que possui o σ mais próximo do seu para que ela seja o seu melhor global na equação de movimento.

$$\sigma = \begin{bmatrix} (f_1^2 - f_2^2)/(f_1^2 + f_2^2 + f_3^2) \\ (f_2^2 - f_3^2)/(f_1^2 + f_2^2 + f_3^2) \\ (f_3^2 - f_1^2)/(f_1^2 + f_2^2 + f_3^2) \end{bmatrix} \quad (16)$$

2.2.2.4 Abordagem multi-objetivo para evolução diferencial

Xue, Sanderson e Graves (2003) propõem o MODE considerando três principais componentes do algoritmo: a mutação, a avaliação baseada em pareto e a seleção. Na mutação, Xue, Sanderson e Graves (2003) define dois tipos de vetores: o diferencial e o de perturbação. Na abordagem mono-objetivo o vetor diferencial é aquele entre o melhor indivíduo e o indivíduo que no momento está sobre a mutação, e seu objetivo é mover os indivíduos em direção ao melhor indivíduo. Para problemas MO, o algoritmo deve encontrar um conjunto de soluções ótimas em vez de apenas uma única solução e, portanto, o vetor

diferencial precisa ser adaptado. Já o vetor de perturbação não é alterado pois os indivíduos são escolhidos aleatoriamente.

Na abordagem de [Xue, Sanderson e Graves \(2003\)](#), um conjunto D é criado a cada geração do algoritmo por meio das operações de avaliação e seleção. Este conjunto mantém as soluções não dominadas da geração atual. Durante a operação de mutação, primeiro é verificado se o indivíduo pertence ao conjunto D , caso não seja, o vetor do melhor indivíduo será o vetor de um dos indivíduos de D escolhido aleatoriamente. Caso o indivíduo pertença a D o vetor diferencial é nulo.

Os indivíduos são avaliados não apenas pelas funções objetivo, mas também por suas dominâncias e fronteiras pareto. De maneira similar ao NSGA-II, o MODE atribui a todos os indivíduos um ranque de acordo com qual fronteira pareto o indivíduo pertence, porem o método adotado para ranquear os indivíduos é o proposto por [Goldberg \(1989\)](#), neste método inicialmente identifica-se os indivíduos não dominados e atribui-se o ranque 1, em seguida estes indivíduos são desconsiderados na próxima iteração e identifica-se os novos indivíduos não dominados e atribui a eles o ranque 2. Esse processo continua até que todos indivíduos sejam ranqueados. MODE também adota o operador de distância de multidão do NSGA-II para comparar indivíduos com o mesmo ranque.

O operador de seleção do MODE escolhe os melhores indivíduos para a próxima geração baseado na aptidão do ranque do indivíduo. Nesta seleção há o parâmetro $\sigma_{multidao}$ que especifica o quanto uma solução deve está próxima de outras soluções, para que ela seja penalizada em sua aptidão. Com isso evita-se que indivíduos muito similares sejam escolhidos e ocorra uma convergência prematura do algoritmo.

2.2.2.5 Multi-objective evolutionary particle swarm optimization (MEPSO)

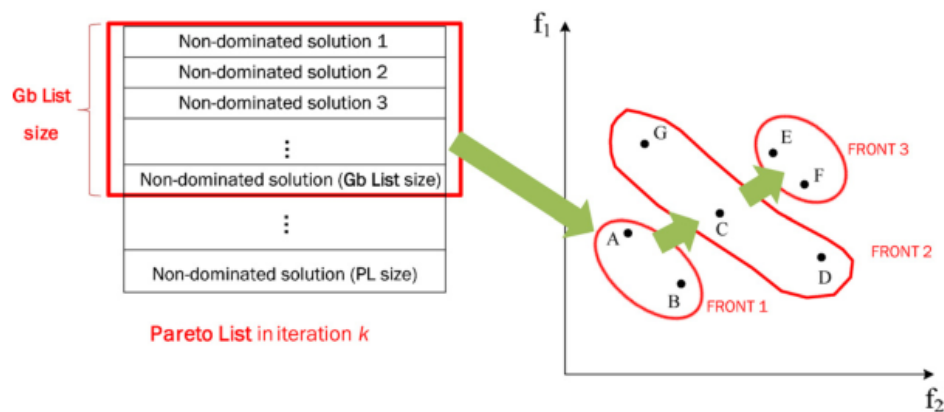
Como seu próprio nome define, o MEPSO de ([MACIEL et al., 2012](#)) é uma versão MO do EPSO que adiciona os mecanismos do NSGA-II para tratar esse tipo de problema. No MEPSO os mecanismos de replicação e mutação não são alterados, enquanto a seleção e avaliação são adaptados utilizando o operador de ordenação por dominância rápida e a seleção baseada na comparação de distância de multidão, provenientes do NSGA-II.

De maneira semelhante a outros algoritmos MO, o MEPSO mantém um arquivo externo denominado de lista pareto. Este arquivo contém o conjunto das soluções não dominadas e é atualizado ao termino de cada iteração do algoritmo. Ao inicio de cada iteração as partículas são ranqueadas e ordenadas de acordo com suas respectivas fronteiras de dominância, i.e. o operador de ordenação por dominância rápida e o calculo da distancia de multidão são aplicados nas partículas do exame.

Ao contrário do EPSO onde existe apenas um único guia de melhor global B_g , no MEPSO cada partícula recebe um melhor global B_g próprio e este é escolhido baseado em

qual fronteira a partícula pertence. MEPSO adota uma estratégia em que o B_g escolhido é uma partícula aleatória da fronteira superior seguinte à fronteira da partícula atual. Como exemplo, as partículas da terceira fronteira utilizam as partículas da segunda fronteira como guias de melhor global. No caso em que a partícula pertença à primeira fronteira, os guias globais serão escolhidos aleatoriamente de um conjunto reduzido de partículas da lista pareto. As partículas da lista pareto são escolhidas de forma a preservar a diversidade, para isto as partículas selecionadas para este conjunto são aquelas com maior distância de multidão. A Figura 9 ilustra este procedimento de escolha para os melhores globais.

Figura 9 – Procedimento de escolha para os melhores globais no MEPSO. As fronteiras da população escolhem os melhores globais baseando-se na fronteira superior seguinte. A primeira fronteira por sua vez escolhe os melhores globais a partir da lista pareto contendo as soluções não dominadas encontradas pelo algoritmo.



Fonte: (MACIEL et al., 2012).

No EPSO a partícula original é comparada com sua replica e aquela que possuir melhor aptidão é selecionada para permanecer no enxame, já no MEPSO a seleção é feita comparando-se todas as partículas primeiramente de acordo com suas fronteiras e em caso de empate de acordo com suas distâncias de multidão.

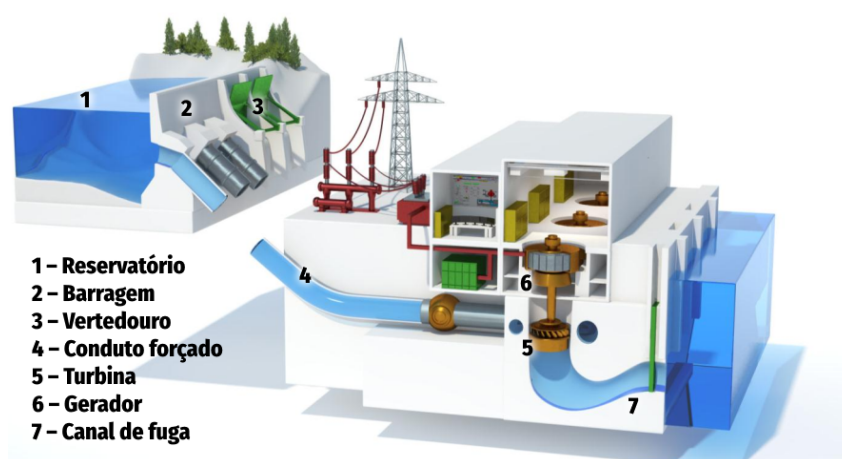
2.3 Usina hidroelétrica

A produção da energia elétrica é o resultado de um processo de transformação das energias potencial e cinética. A energia potencial da água armazenada no reservatório é transformada pela turbina em energia cinética que, através de um eixo, é transmitida ao gerador. No gerador, a energia cinética é transformada em energia elétrica que, após passar por uma subestação elevadora de tensão, é injetada no sistema de transmissão para entregá-la aos centros de consumo (ARCE, 2006).

2.3.0.1 Principais componentes de uma UHE

Os principais componentes da estrutura de uma UHE são: reservatório, barragem, vertedouro, tomadas d'água, condutos forçados, conjuntos de turbina-gerador e o canal de fuga. Alguns destes componentes são mostrados na [Figura 10](#).

Figura 10 – Corte lateral de uma UHE e seus os principais componentes.



Fonte: Adaptado de [Fernandes \(2015\)](#).

A barragem é uma barreira artificial construída geralmente em terra e concreto armado em cursos d'água, responsável por reter grandes quantidades de água formando um lago artificial, denominado reservatório. Localizado na barragem a tomada d'água é a estrutura ou o local cuja finalidade é controlar, regular, derivar e receber a água oriunda do reservatório. Também localizado na barragem o vertedouro consiste basicamente em uma passagem, controlada por comportas, cuja função é escoar o excesso de água, descarregando-a para a jusante de forma segura.

Os reservatórios de água são unidades hidráulicas de acumulação e passagem de água, para atender a quantidade de água necessária para geração, bem como a vazão ou o escoamento da mesma quando necessário. O reservatório possui dois níveis de operação, a cota mínima e a cota máxima. Quando o volume de água é inferior à cota mínima, este volume é chamado de volume morto e não há geração de energia. A diferença do volume na cota máxima e o volume morto define o volume útil do reservatório.

A água que passa pela tomada d'água é levada através dos condutos forçados até a casa de força. A casa de força aloja os equipamentos de outros componentes como a caixa espiral, os conjuntos de turbina-gerador, o sistema de excitação e o regulador de velocidade.

O conjunto turbina-gerador é responsável pela geração da energia elétrica. As turbinas hidráulicas usadas nas usinas podem ser de dois tipos: turbinas de reação e turbinas de ação. De acordo com [Schreiber \(1981\)](#), a turbina hidráulica de reação é aquela

que o trabalho mecânico é obtido pela transformação da energia cinética e de pressão da água, enquanto as de ação transforma apenas a energia cinética. Após sua passagem pelas turbinas, a água é restituída ao leito natural do rio através do canal de fuga.

A turbina é um equipamento composto de vários componentes. Dentre eles, destacam-se o distribuidor, o rotor e a pá. O distribuidor é o órgão fixo constituído por pás moveis em torno do seu eixo que formam canais, através dos quais se conduz a vazão turbinada para o rotor. O rotor da turbina é o órgão giratório sobre o qual age a água que foi conduzida a ele pelo distribuidor.

2.3.1 Balanço hídrico de um reservatório de uma UHE

Em hidrologia, o balanço hídrico é uma quantificação da água que entra e sai de um sistema em uma determinada variação de tempo. Para uma UHE o balanço hídrico de seu reservatório é fundamental para suas operações, pois é pelo balanço hídrico que se define a evolução do volume e das cotas do reservatório.

As principais variáveis intervenientes do balanço hídrico de um reservatório de uma UHE são:

- A vazão afluente, que é a quantidade de água que chega ao reservatório pelo fluxo da bacia hidrográfica ligada ao reservatório e de aproveitamentos próximos ao montante da UHE;
- A evaporação, que é a quantidade de água retirada da superfície da lamina de água do reservatório devido à radiação solar;
- A vazão turbinada, que é a quantidade de água que passa pelas turbinas para a geração de energia pela a UHE;
- A vazão vertida, que é a quantidade de água que é liberada do reservatório pelos vertedouros.

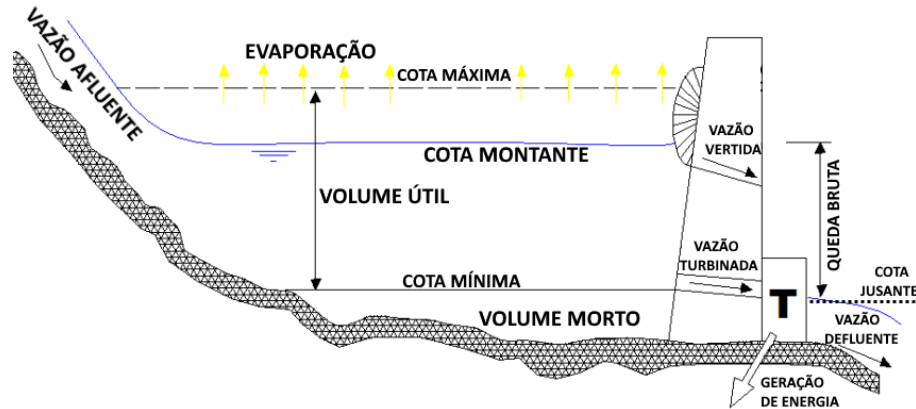
A combinação da vazão vertida e da vazão turbinada define a vazão defluente que é a quantidade de água que sai das estruturas hidráulicas da UHE. O volume V do reservatório de uma UHE em um instante t é definido pela [Equação \(17\)](#) e ilustrado pela [Figura 11](#).

$$V_t = V_{t-1} + Qa_{t-1} - Qt_{t-1} - Qv_{t-1} - E_{t-1} \cdot A_{t-1}, \quad (17)$$

na qual,

- V é o volume do reservatório;
- Qa é a vazão afluente;
- Qt é a vazão turbinada;
- Qv é a vazão vertida;

Figura 11 – Principais variáveis e cotas envolvidas no balanço hídrico do reservatório de uma UHE.



Fonte: Adaptado de BRAVO (2006).

- E é a evaporação líquida;
- A é a área ocupada pela água no reservatório.

As unidades das variáveis são usualmente: hm^3 para o volume, $m^3 \cdot s^{-1}$ para as vazões, mm por dia para a evaporação e km^2 para a área ocupada pela água. De acordo com a variação de tempo, é necessário adicionar uma constante de conversão para as variáveis em que tempo de sua unidade não corresponde à ordem de grandeza da variação do tempo.

Conforme ilustrado na Figura 11 a cota montante é definida pelo volume do reservatório, já a cota jusante é dependente dos valores da vazão defluente. Com a diferença da cota montante para a cota jusante tem-se a altura da queda bruta que tem influência direta e proporcionalmente na geração de energia. Segundo Finardi (2003), para as UHE brasileiras em geral, as cotas montante e jusante podem ser definidas por um polinômio de quarta ordem.

Para a cota montante f_{cm} , dado um volume em hm^3 , o seu valor em m é calculado pelo polinômio da equação Equação (18),

$$f_{cm} = a_0 + a_1 \cdot V + a_2 \cdot V^2 + a_3 \cdot V^3 + a_4 \cdot V^4. \quad (18)$$

Para a cota jusante f_{cj} , dado uma vazão defluente em $m^3 s^{-1}$, o seu valor em m é calculado pelo polinômio da equação Equação (19),

$$f_{cj} = b_0 + b_1 \cdot (Qt + Qv) + b_2 \cdot (Qt + Qv)^2 + b_3 \cdot (Qt + Qv)^3 + b_4 \cdot (Qt + Qv)^4. \quad (19)$$

Logo, a queda bruta hb é definida como $hb = f_{cm} - f_{cj}$. Os coeficientes $a_0 \cdots a_4$

e $b_0 \dots b_4$ geralmente são obtidos através de experimentos e regressões sobre a curva do comportamento da respectiva cota com o volume ou vazão defluente.

2.3.2 Modelo matemático para o despacho elétrico de uma UHE

Um dos estudos de caso tratado nesse trabalho envolve a geração de energia elétrica por UHE. O modelo matemático de despacho elétrico utilizado como base foi proposto por [Marcelino \(2017\)](#). Esta seção apresenta tal modelo de maneira abreviada.

O objetivo deste modelo é maximizar a eficiência da produção de energia elétrica da UHE e esta é dada pela razão da potência gerada pela usina com a vazão de água utilizada pelas unidades geradoras. A produção de energia de uma unidade geradora é descrita pela equação 20:

$$ph_{jt} = gk \cdot [\rho_{0j} + \rho_{1j}hl_{jt} + \rho_{2j}q_{jt} + \rho_{3j}hl_{jt}q_{jt} + \rho_{4j}hl_{jt}^2 + \rho_{5j}q_{jt}^2] \cdot [Hb_{jt} - \Delta_{Hjt}] \cdot q_{jt}, \quad (20)$$

na qual,

- ph_{jt} é a potência gerada pela unidade geradora j no tempo t em MW ;
- g é a aceleração da gravidade $9,8m/s$;
- $k = 10^3kg/m$ uma constante que converte cavalo vapor, HP , em MW ;
- $\rho_{0j} \dots \rho_{5j}$ são os coeficientes operativos da unidade geradora que foram obtidos por regressão linear multivariável e estão dispostos na Tabela 1;
- Hb_{jt} é a altura de queda bruta na unidade geradora j no tempo t em m ;
- hl_{jt} é a altura de queda líquida na unidade geradora j no tempo t em m ;
- q_{jt} é a vazão turbinada na unidade geradora j no tempo t em m^3/s ;
- Δ_{Hjt} são as perdas totais referentes ao atrito nos condutos forçados na unidade geradora j no tempo t em m .

Tabela 1 – Coeficientes Operativos.

Coeficiente	Valor
ρ_{0j}	1,4630e-01
ρ_{1j}	1,8076e-02
ρ_{2j}	5,0502e-03
ρ_{3j}	-3,5254e-03
ρ_{4j}	-1,1234e-03
ρ_{5j}	1,4507e-05

Fonte: [Marcelino \(2017\)](#).

Os termos q_{jt} são as variáveis de decisão do modelo, com isto pode-se definir a

função objetivo do modelo de acordo com a equação 21, no qual $x = [q_{1t}, q_{2t}, q_{2t}, \dots, q_{jt}]$:

$$\text{Maximize } F_1(x) = \frac{\sum_{j=1}^{J(r)} ph_{jt}}{\sum_{j=1}^{J(r)} q_{jt}}, \quad (21)$$

a função objetivo está sujeita às seguintes restrições:

$$\sum_{j=1}^{J(r)} ph_{jt} = Dm,$$

$$q_{jtmin} \leq q_{jt} \leq q_{jtmax},$$

$$ph_{jk}^{min} \sum_{k=1}^{\emptyset_j} Z_{jk} \leq ph_{jt} \leq ph_{jk}^{max} \sum_{k=1}^{\emptyset_j} Z_{jk},$$

$$Z_{jk} \in \{0,1\}, \sum_{k=1}^{\emptyset_j} Z_{jk} \leq 1.$$

O [Quadro 1](#) descreve os demais parâmetros utilizados no modelo, as equações de restrição representam as restrições operacionais da UHE. A primeira restrição define que energia elétrica gerada deve ser igual a demanda solicitada para a UHE. A segunda restrição define que a vazão calculada deve respeitar os limites mínimos e máximos das capacidades das unidades geradoras. A terceira restrição define que a potência gerada deve respeitar os limites mínimo e máximo das unidades geradoras. Por fim a quarta restrição nos informa que existe apenas uma zona de operação para as unidades geradoras, i.e., a unidade permanece ligada ou desligada durante todo o processo.

Quadro 1 – Descrição dos parâmetros utilizados no modelo.

Parâmetro	Significado
ph_{jtmin}	Potência mínima
ph_{jtmax}	Potência máxima
Dm	Demanda de energia solicitada
q_{jt}	Vazão de água da unidade j no tempo t
q_{jtmin}	Vazão de água mínima
q_{jtmax}	Vazão de água máxima
Z_{jk}	Estado da turbina j no tempo k

Fonte: [Marcelino \(2017\)](#).

Com este modelo e a aplicação de algoritmos evolucionários, [Marcelino \(2012\)](#) demonstrou que a usina usada como estudo de caso pode operar de maneira mais eficiente

criando um modo de controle otimizado. Neste novo modo de operação a distribuição das vazões de água é feita de maneira distinta para cada unidades geradoras.

Um segundo objetivo foi proposto por [Marcelino et al. \(2015\)](#), este novo objetivo reduz a distância entre as vazões otimizadas das unidades geradoras e as vazões do modo controle conjunto, que é a o modo de operação padrão da UHE. No modo controle conjunto todas as unidades geradoras recebem a mesma vazão de água. A motivação para criação deste novo objetivo é aumentar a confiança da equipe de operadores da UHE no modo controle otimizado, mostrando que existem vazões otimizadas que são próximas das vazões utilizadas no modo controle conjunto.

O segundo objetivo é representado pela função na [Equação \(22\)](#), em que o termo q_{cc} é a vazão de água para cada unidade geradora no modo controle conjunto. Não há alterações no restante do modelo anterior para tratar o novo objetivo. Portanto, as equações do modelo descrito anteriormente também são aplicadas no modelo multiobjetivo. Com isto o modelo multiobjetivo é definido a seguir:

$$\begin{aligned} \text{Maximize } F_1(x) &= \frac{\sum_{j=1}^{J(r)} ph_{jt}}{\sum_{j=1}^{J(r)} q_{jt}}, \\ \text{Minimize } F_2(x) &= \sqrt{\sum_{j=1}^{J(r)} (q_{jt} - q_{cc})^2}. \end{aligned} \quad (22)$$

sujeito às seguintes restrições:

$$\begin{aligned} \sum_{j=1}^{J(r)} ph_{jt} &= Dm, \\ q_{jtmin} &\leq q_{jt} \leq q_{jtmax}, \\ ph_{jk}^{min} \sum_{k=1}^{\emptyset_j} Z_{jk} &\leq ph_{jt} \leq ph_{jk}^{max} \sum_{k=1}^{\emptyset_j} Z_{jk}, \\ Z_{jk} &\in \{0,1\}, \sum_{k=1}^{\emptyset_j} Z_{jk} \leq 1. \end{aligned}$$

3 Metodologia

Para Gil (2010), uma pesquisa científica pode ser classificada quanto aos objetivos e aos procedimentos. Para os objetivos, as pesquisas podem ser exploratórias, descritiva e/ou explicativa.

Este trabalho tem natureza exploratória pois visa aprimorar o C-DEEPSO para um contexto MO afim de corroborar com a hipótese de Marcelino et al. (2018) de que algoritmos híbridos melhoram a eficiência da busca. Também é escopo deste trabalho o aprimoramento do modelo matemático de Marcelino (2012), Marcelino (2017) para tratar a dinâmica de UHEs em cascata.

Quanto aos procedimentos Gil (2010), classifica a pesquisa nas seguintes categorias: bibliográfica, documental, experimental, levantamento, estudo de campo e estudo de caso. Para este trabalho foram empregados os seguintes procedimentos:

1. A pesquisa bibliográfica, que é usualmente parte das pesquisas de objetivo exploratório. A bibliografia pesquisada deste trabalho, visa analisar e identificar técnicas e algoritmos MO para adaptar o C-DEEPSO e trabalhos que abordam geração de energia de UHEs em cascata como estudo de caso, para a proposição de um novo modelo.
2. A pesquisa experimental foi utilizada neste trabalho para o desenvolvimento e validação do algoritmo híbrido MO baseado no C-DEEPSO, identificando os melhores operadores e procedimentos para o algoritmo através de experimentos com funções *benchmark* MO. Esta pesquisa também será aplicada aos estudos de casos, sendo um deles a validação do modelo proposto.
3. E por fim o estudo de caso, que foi utilizado para avaliar o desempenho do algoritmo MO híbrido desenvolvido neste trabalho em problemas, sendo um destes problemas o modelo proposto.

A metodologia para o cumprimento dos objetivos do trabalho é descrita pelas etapas a seguir:

1. Revisão da bibliográfica: visa identificar e analisar trabalhos com algoritmos semelhantes ao C-DEEPSO e outros algoritmos no contexto MO, para que estes sirvam de guia para a criação do novo algoritmo híbrido MO. Uma outra revisão é relativa aos estudos de casos, feita sobre os trabalhos de UHEs com foco em modelos que tratam a dinâmica de usinas em cascatas, pretende-se incorporar as ideias destes

modelos com o modelo de [Marcelino \(2017\)](#) para que este novo modelo trate usinas em cascata.

2. Desenvolvimento de um novo algoritmo híbrido MO baseado nos fundamentos do C-DEEPSO: nesta etapa os mecanismos e operadores dos algoritmos analisados na revisão bibliográfica são avaliados para identificar aqueles que podem ser melhor incorporados ao funcionamento do C-DEEPSO e com estes será criado um novo algoritmo híbrido MO fundamentado no funcionamento do C-DEEPSO.
3. Validação do novo algoritmo: realização de experimentos e análise estatística com o algoritmo proposto na etapa anterior e os algoritmos NSGA-II e SPEA2 utilizando funções *benchmark*.
4. Proposta de modelo MO para UHE em cascata: de maneira semelhante à etapa 2, um novo modelo será proposto utilizando o modelo de [Marcelino \(2017\)](#) como base e incorporando ideias de outros modelos levantados pela revisão bibliográfica visando criar um modelo MO e com a dinâmica da geração de energia de UHEs em cascata.
5. Avaliação do algoritmo proposto com os estudos de caso: realização de experimentos com o algoritmo proposto, o NSGA-II e o SPEA2 sobre o modelo proposto de UHEs em cascata e o modelo MO de [Marcelino et al. \(2015\)](#). Após a conclusão dos experimentos, os resultados serão avaliados por meio de análise estatística com o intuito de verificar se o algoritmo proposto apresenta performance satisfatória para problemas reais.

3.1 Planejamento de experimentos

Nesta seção serão apresentados brevemente os experimentos planejados para a realização deste trabalho e também as métricas e técnicas estatísticas para a análise dos resultados dos experimentos.

3.1.1 Experimentos planejados

Conforme apresentado nas etapas da metodologia, os experimentos deste trabalho podem ser separados em três etapas:

1. Validação de diferentes configurações do algoritmo proposto: este experimento tem como objetivo identificar quais as melhores configurações de parâmetros e operadores do algoritmo proposto apresentam melhores resultados. Para cumprir o objetivo serão aplicados testes de desempenho com funções *benchmark* com o algoritmo proposto
2. Validação do algoritmo proposto com outros algoritmos clássicos em *benchmarks*: avaliar o desempenho do algoritmo proposto com as melhores configurações encon-

tradas na etapa anterior, comparando-o com os algoritmos NSGA-II e SPEA2 em funções *benchmark*.

3. Validação do modelo de UHE em cascata e do algoritmo proposto em problemas reais: realização de simulações com o algoritmo proposto, o NSGA-II e o SPEA2 aplicados ao modelo desenvolvido para estudo de caso. Objetivo deste experimento é verificar o desempenho do algoritmo proposto em problemas reais e validar o modelo de UHEs em cascata.

Dado a natureza estocástica de todos os algoritmos envolvidos, para cada teste de desempenho haverá pelo menos 30 execuções de cada algoritmo para que seus resultados tenham melhor significância estatística. O valor 30 se baseia no interesse prático, dado que a aproximação normal será satisfatória para amostragens maiores do que 30. O teorema do limite central funciona se uma distribuição populacional não é muito diferente da normal se a quantidade de amostragens é menor do que 30 (MONTGOMERY, 2004). As métricas e os métodos estatísticos para avaliar os resultados destes experimentos serão detalhados nas seções seguintes.

3.1.2 Métricas de comparação de algoritmos multiobjetivo

Segundo Zitzler (1999) os objetivos de um algoritmo que otimiza problemas MO são:

- Minimizar a distância entre a fronteira de soluções não-dominadas encontrada pelo algoritmo e o pareto ótimo do problema.
- Uma boa distribuição (na maioria dos casos uniforme) de soluções encontradas é desejável.
- Maximizar a extensão da fronteira não-dominada encontrada pelo algoritmo, i.e., para cada objetivo uma extensa variedade de valores deve ser coberta pelas soluções não-dominadas do algoritmo.

Ao longo dos anos, diversas métricas foram propostas para comparação de algoritmos MO baseando-se nesses três objetivos e de acordo o *review* de Riquelme, Lücken e Baran (2015) as métricas mais utilizadas em ordem decrescente são: hipervolume, distância geracional (generational distance), indicador épsilon e a distância geracional invertida (inverted generational distance).

Neste trabalho as métricas escolhidas são a de hipervolume ou S-metric e a distância geracional. O primeiro objetivo será quantificado pela distância geracional, enquanto os outros dois objetivos são quantificados pela S-metric. É importante ressaltar que o primeiro objetivo só é possível de ser avaliado se o pareto ótimo do problema é conhecido,

portanto o uso da distância geracional é viável somente nos primeiros experimentos com funções *benchmark*.

A distância geracional foi introduzida por Veldhuizen e Lamont (1998) para estimar a distância entre os elementos da fronteira de soluções não-dominadas daqueles na fronteira pareto ótimo, seu cálculo é definido pelo equação 23 :

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n}, \quad (23)$$

no qual n é a quantidade de elementos na fronteira encontrada pelo algoritmo, d_i é a distância euclidiana (medida no espaço objetivo) entre cada solução do algoritmo com a solução mais próxima na fronteira pareto. Nesta equação menores valores de GD indicam maior proximidade entre as duas fronteiras, sendo $GD = 0$ indicador de que todos os elementos encontrados pelo algoritmo estão no pareto ótimo.

Originalmente proposto por Zitzler e Thiele (1998), a S-metric pode ser interpretada segundos seus próprios autores como o tamanho do espaço coberto ou o tamanho do espaço dominado. O cálculo da S-metric de um conjunto U de soluções é dado pela equação:

$$S(U) = L\left(\bigcup_{u \in U} \{x | u < x < x_{ref}\}\right), \quad (24)$$

no qual L é a medida de Lebesgue e x_{ref} é um ponto de referencia que é dominado por todas as soluções de U . Assim o S-metric é a união dos hipercubos formados por cada solução de U e o ponto de referência x_{ref} .

Uma nova métrica foi proposta para este trabalho, no caso de não ser possível definir uma diferença estatística entre os algoritmos com as métricas anteriores. Como um critério de desempate, será medido o número de avaliações na função de aptidão até que o algoritmo encontre um pareto com um hipervolume próximo a um limiar definido de 10^{-4} . Desta forma, verifica-se qual algoritmo é mais rápido computacionalmente para atingir este limiar.

3.1.3 Métodos para análise estatística

Os resultados dos experimentos serão avaliados seguindo uma sequência de três métodos que são: o *boxplot*, a análise de variância (ANOVA) e o teste de Tukey (MONTGOMERY, 2012). Esta sequência foi organizada de maneira que caso a comparação de um método não seja conclusiva, o método seguinte tenta esclarecer a incerteza anterior.

O *boxplot* ou grafico de caixa é um teste não paramétrico que apresenta características de um conjunto de dados números através de seu valor mínimo, valor máximo, quartis inferior, centralizado e superior em um gráfico em formato de caixa. O *boxplot* permite uma

comparação rápida entre diferentes conjuntos de dados em relação à dispersão de dados, mediana, valores máximos e mínimos, em situações em que não há sobreposição das caixas é possível afirmar que um conjunto de dados é diferente de outro.

A ANOVA é um método estatístico que permite avaliar hipóteses sobre as médias e variâncias dos conjuntos de dados. Desta forma, visa verificar se há diferenças significativas entre as médias e se os fatores exercem influência em alguma variável dependente (MONTGOMERY, 2012). As hipóteses avaliadas são definidas formalmente como:

$$\begin{cases} H_0 : \mu_i = \mu_j, \forall i, j, \\ H_1 : \mu_i \neq \mu_j, \text{ para algum } i. \end{cases}$$

Nesta formulação a hipótese H_0 é a hipótese nula e pressupõe a igualdade das médias e já a hipótese alternativa H_1 indica a desigualdade das médias. O resultado do ANOVA é dado pela medida P-valor que estabelece o menor nível de significância que leva à rejeição de H_0 o que caracteriza a existência de pelo menos uma média que é diferente das demais.

Apesar da ANOVA ser um método mais robusto que o *boxplot* para verificar a existência de diferenças nas médias, ele não é capaz de especificar onde esta diferença se encontra, i.e., a ANOVA não consegue afirmar qual conjunto é superior ao outro. Para solucionar esta questão aplica-se o teste de Tukey, que é um método estatístico que pode ser interpretado como a comparação das médias entre os diferentes grupos, com a variância entre todos os indivíduos dentro desses grupos. Com o teste de Tukey podemos identificar as diferenças entre os conjuntos e partir dele classificar os conjuntos.

3.1.4 Pareto combinado

O pareto combinado é formado pelas melhores soluções encontradas por diferentes execuções de um algoritmo. As soluções finais das execuções são unidas em uma mesma população, em seguida elimina-se as soluções dominadas e aplica-se um operador de diversidade para se obter um pareto de tamanho igual ao das populações usadas nas execuções do algoritmo.

Com o pareto combinado é possível fazer uma avaliação empírica de diferentes algoritmos. Como exemplo, no caso do pareto combinado de algoritmo dominar o pareto combinado de outro, espera-se que desempenho do primeiro algoritmo seja superior ao do segundo. Entretanto, uma análise por paretos combinados não possui relevância estatística suficiente para confirmar o desempenho do algoritmo em relação a outro, sendo assim necessário o uso dos métodos estatísticos definidos anteriormente.

3.1.5 Funções *benchmark*

Zitzler, Deb e Thiele (2000) definiram um conjunto de seis funções testes para avaliar o desempenho de otimizadores multiobjetivo, em referencia aos nomes de seus autores as funções comumente chamadas de ZDT. Neste trabalho a ZDT1, ZDT2, ZDT3, ZDT4 e ZDT6 são as funções escolhidas como *benchmark* e serão detalhadas a seguir. ETHZ (2019) fornece valores de referência para os paretos ótimos das funções ZDT, estes valores são aproximações feitas de acordo com uma quantidade finita de pontos, distribuídos de forma a gerar o maior hipervolume possível.

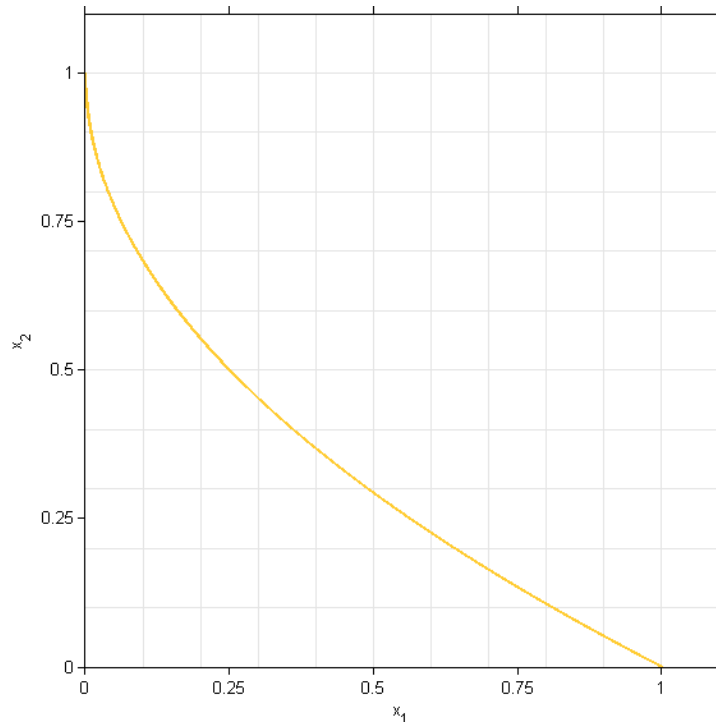
3.1.5.1 ZDT1

A ZDT1 possui um pareto ótimo convexo e é definida pela seguinte formulação:

$$\begin{aligned} F1 &= x_1, \\ F2 &= G(x) \cdot (1 - \sqrt{F1/G(x)}), \\ G(x) &= 1 + (9/n - 1) \sum_{i=2}^n x_i, \\ 0 &\leq x_i \leq 1, i = 1, \dots, n, \end{aligned} \tag{25}$$

o valor de n sugerido pelos autores é 30. Seu pareto ótimo é gerado com $x_2 = 1 - \sqrt{x_1}$, a Figura 12 apresenta o gráfico deste pareto ótimo. Usando o ponto de referencia (11 11) o hipervolume máximo para 50 pontos é de 120,657446.

Figura 12 – Pareto ótimo para a ZDT1.



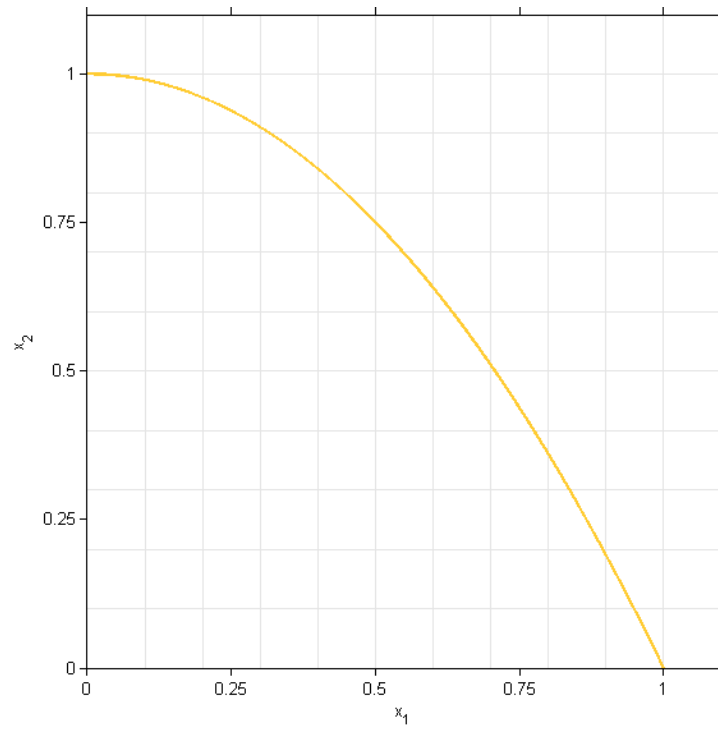
3.1.5.2 ZDT2

A ZDT2 possui um pareto ótimo não convexo e é definida pela seguinte formulação:

$$\begin{aligned}
 F1 &= x_1, \\
 F2 &= G(x) \cdot (1 - (F1/G(x))^2), \\
 G(x) &= 1 + (9/n - 1) \sum_{i=2}^n x_i, \\
 0 &\leq x_i \leq 1, i = 1, \dots, n,
 \end{aligned} \tag{26}$$

o valor de n sugerido pelos autores é 30. Seu pareto ótimo é gerado com $x_2 = 1 - x_1^2$, a [Figura 13](#) apresenta o gráfico deste pareto ótimo. Usando o ponto de referencia (11 11) o hipervolume máximo para 50 pontos é de 120,324398.

Figura 13 – Pareto ótimo para a ZDT2.



3.1.5.3 ZDT3

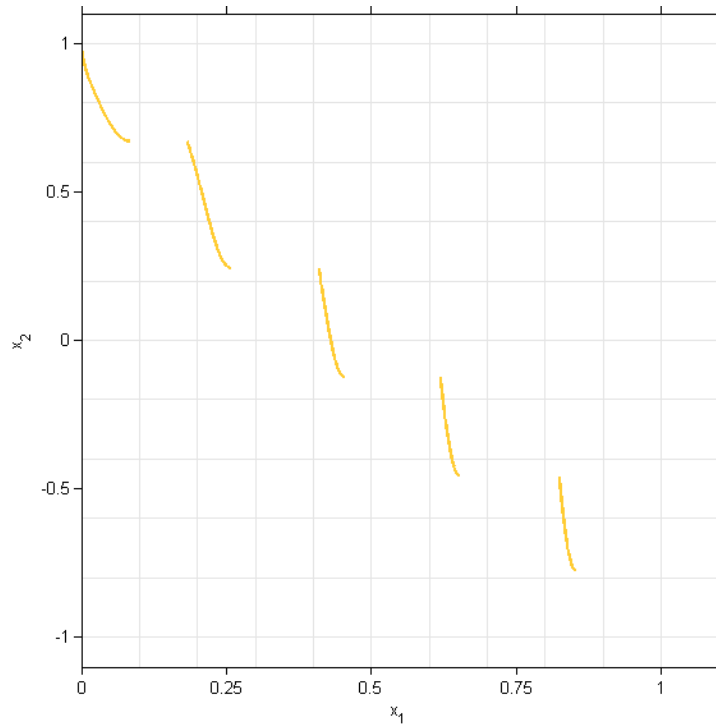
A ZDT3 possui um pareto ótimo composto por partes não contínuas e convexas, apesar da descontinuidade no pareto o espaço das variáveis de decisão é contínuo. A ZDT3 é definida pela seguinte formulação:

$$\begin{aligned}
 F1 &= x_1, \\
 F2 &= G(x) \cdot (1 - \sqrt{F1/G(x)} - (F1(x)) \text{sen}(10\pi F1), \\
 G(x) &= 1 + (9/n - 1) \sum_{i=2}^n x_i, \\
 0 &\leq x_i \leq 1, i = 1, \dots, n,
 \end{aligned} \tag{27}$$

o valor de n sugerido pelos autores é 30. Seu pareto ótimo é gerado com $x_2 = 1 - \sqrt{x_1} - x_1 \text{seno}(10\pi x_1)$ com $x_1 \in P$, $P = [0 \ 0,0830015349] \cup (0,1822287280 \ 0,2577623634] \cup (0,4093136748 \ 0,4538821041] \cup (0,6183967944 \ 0,6525117038] \cup (0,8233317983 \ 0,8518328654]$.

A Figura 14 apresenta o gráfico deste pareto ótimo. Usando o ponto de referencia (11 11) o hipervolume máximo para 50 pontos é de 128,773637.

Figura 14 – Pareto ótimo para a ZDT3.



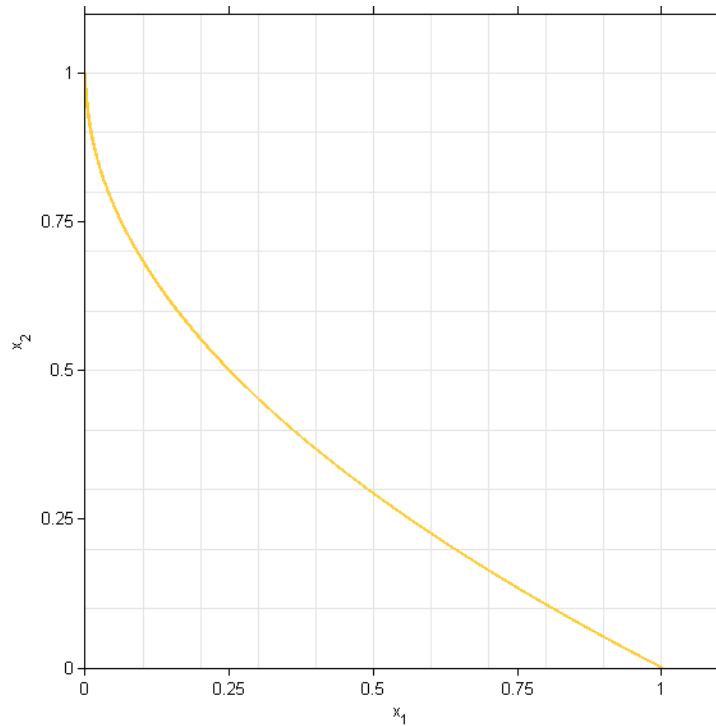
3.1.5.4 ZDT4

A ZDT4 possui um pareto ótimo convexo, porem apresenta 21^9 paretos locais, portanto a ZDT4 visa avaliar a capacidade do otimizador em lidar com a multimodalidade. A ZDT4 é definida pela seguinte formulação:

$$\begin{aligned}
 F1 &= x_1, \\
 F2 &= G(x) \cdot (1 - (F1(x))^2), \\
 G(x) &= 1 + 10(n - 1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)), \\
 0 &\leq x_1 \leq 1, \\
 -5 &\leq x_i \leq 5, i = 1, \dots, n,
 \end{aligned} \tag{28}$$

o valor de n sugerido pelos autores é 10. Seu pareto ótimo é gerado com $x_2 = 1 - \sqrt{x_1}$. A [Figura 15](#) apresenta o gráfico deste pareto ótimo. Usando o ponto de referencia (11 11) o hipervolume máximo para 50 pontos é de 120,657446.

Figura 15 – Pareto ótimo para a ZDT4.



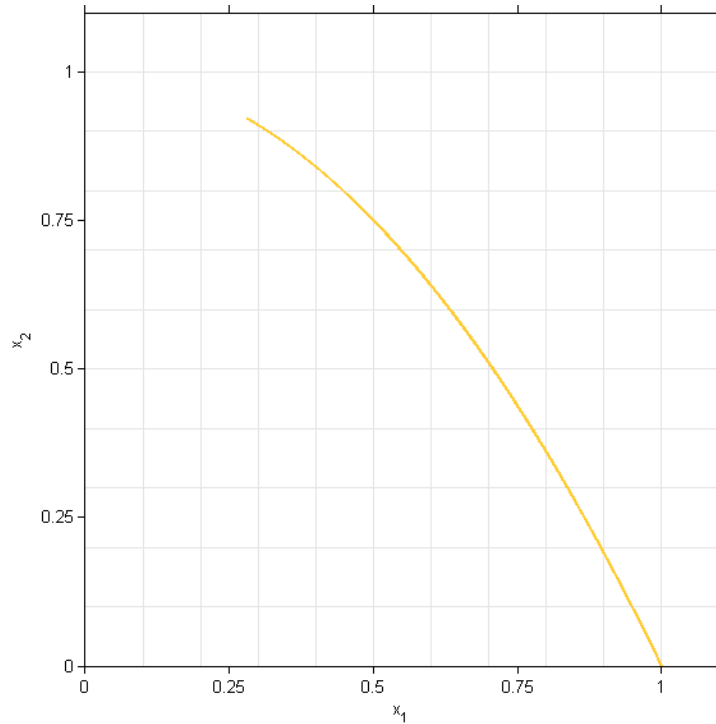
3.1.5.5 ZDT6

A ZDT6 possui um pareto ótimo não convexo e se destaca pela não uniformidade de suas soluções. A ZDT6 é definida pela seguinte formulação:

$$\begin{aligned}
 F1 &= 1 - \epsilon^{-4x_1} \cdot \sin^6(6\pi x_1), \\
 F2 &= 1 - (F1(x))^2, \\
 G(x) &= 1 + 9\left[\left(\sum_{i=2}^n x_i\right)/(n-1)\right]^{0,25}, \\
 0 &\leq x_i \leq 1, i = 1, \dots, n,
 \end{aligned} \tag{29}$$

o valor de n sugerido pelos autores é 10. Seu pareto ótimo é gerado com $x_2 = 1 - \sqrt{x_1}$ e $x_1 \in [0,2807753191 \ 1]$. A [Figura 16](#) apresenta o gráfico deste pareto ótimo. Usando o ponto de referencia (11 11) o hipervolume máximo para 50 pontos é de 117,511658.

Figura 16 – Pareto ótimo para a ZDT6.



4 Desenvolvimento do algoritmo híbrido multiobjetivo MESH

O C-DEEPSO é um algoritmo mono-objetivo híbrido que possui um desempenho competitivo na otimização de diferentes problemas reais, tais como: despacho elétrico em UHE, sistema *microgrid* híbrido e fluxo de potência ótimo. Devido a esse desempenho, motivou-se o desenvolvimento do *Multi-objective Evolutionary Swarm Hybrid* (MESH), um novo algoritmo híbrido que se fundamenta no C-DEEPSO, porém com a capacidade de tratar problemas MO.

Apesar do MESH seguir a motivação anterior, seu desenvolvimento foi além de modificações e adições para lidar com o contexto MO. Também foram feitas outras alterações em alguns fundamentos C-DEEPSO melhorando-se o desempenho do novo algoritmo.

4.1 *Multi-objective evolutionary swarm hybrid* (MESH)

O MESH mantém o movimento de partículas original do C-DEEPSO, logo sua formulação segue a [Equação \(30\)](#). O cálculo do termo X_{st} por sua vez se tornou um novo operador no MESH, que se fundamenta na mutação do DE com ajustes para tratar o contexto MO. O diagrama do funcionamento do MESH é apresentado pela [Figura 17](#) ao final desta seção. Maiores detalhes das alterações, adições e operadores do algoritmo são apresentados nos parágrafos seguintes.

$$\mathbf{V}_i^{\text{novo}} = \mathbf{W}_I^* \mathbf{V}_i + \mathbf{W}_A^* (\mathbf{X}_{st} - \mathbf{X}_i) + \mathbf{W}_C^* \mathbf{C}(\mathbf{B}_g^* - \mathbf{X}_i), \quad (30)$$

Assim como os algoritmos NSGA-II e SPEA2, o MESH adota a estratégia de paretos e dominância de soluções. Esta estratégia gera soluções mais completas e de maior qualidade em relação a outras estratégias, como por exemplo linearização das funções objetivo. Em MESH, o operador de ordenação por dominância rápida do NSGA-II é utilizado para classificar as partículas da população em fronteiras. Estas fronteiras são utilizadas durante os processos de escolha das partículas guias, de atualização da memória e seleção das partículas da população que serão usadas na próxima iteração do algoritmo.

A memória no MESH passa a ser um conjunto que mantém as partículas com soluções não dominadas encontradas pelo algoritmo e permanece como um conjunto separado da população. Portanto, uma partícula pertencente à memória será removida somente se esta for dominada por uma partícula da população.

A atualização da memória é feita combinando-se a primeira fronteira da população com a memória em um mesmo conjunto. Em seguida, o operador de ordenação do NSGA-II é aplicado a este conjunto. Como o objetivo é se identificar as partículas com soluções não dominadas. O operador é interrompido quando a primeira fronteira é formada e então se obtém a nova memória. Caso a fronteira encontrada seja maior que o tamanho do máximo da memória, aplica-se o operador de distância de multidão.

Conforme discutido anteriormente na fundamentação teórica, a escolha das partículas guias no PSO em um contexto MO não é trivial como em um contexto mono-objetivo. Para o MESH, adotou-se a seguinte nomenclatura para estas partículas:

- Guia individual (G_i), corresponde ao conjunto de melhores soluções encontradas pela partícula. Equivalente a partícula B_i utilizada no termo de memória do PSO.
- Guia de enxame (G_s), corresponde a uma solução encontrada pelo exame que é superior à atual solução da partícula. Está solução é proveniente de uma outra partícula da população atual ou da memória. Equivalente a partícula B_g utilizada no termo de cooperação do PSO.

Devido ao conceito de dominância, é possível que em um conjunto de soluções não haja dominância entre as soluções. Devido a isto, no MESH uma partícula pode conter um conjunto de guias individuais em vez de uma única guia individual. Sempre que uma partícula gera uma nova solução, esta solução é comparada com as demais soluções que se encontram atualmente como guias individuais. Se a nova solução não for dominada por nenhuma outra solução do conjunto de guias individuais, a nova solução é adicionada ao conjunto. Caso a nova solução domine alguma outra solução do conjunto, a solução dominada é removida do conjunto de guias individuais.

A escolha do guia de enxame é realizada por um operador e é feita para cada partícula. Este operador adota o método de proximidade com uma outra partícula de acordo com o valor de σ , como proposto por [Mostaghim e Teich \(2003\)](#) no MOPSO. O método sigma pode ser aplicado à apenas a memória, de maneira que cada partícula encontra uma outra partícula na memória que seja a mais próxima.

Outra alternativa é combinar o método sigma com o procedimento de escolha de melhor global similar ao MEPSO de [Maciel et al. \(2012\)](#). Nesta estratégia o guia de enxame de uma partícula é a partícula mais próxima que se encontra na fronteira superior seguinte à atual da partícula. Se a fronteira atual da partícula for a primeira fronteira, a escolha será feita com a memória. A [Figura 18](#), ao final desta seção, exemplifica as duas opções para o operador de escolha do guia de enxame.

O cálculo do termo X_{st} no C-DEEPSO foi renomeado no MESH como um operador de mutação diferencial. Neste operador, cada partícula gera uma nova solução baseando-se em uma das estratégias evolutivas do DE. A nova solução gerada sofre um cruzamento

parcial com um guia individual ou guia de enxame de acordo uma taxa de cruzamento definida nos pesos W do algoritmo. No caso de haver dominância da nova solução sobre a atual, esta substituirá a solução atual da partícula e verifica-se há necessidade de atualizar o guia individual.

No MESH, os vetores utilizados na mutação diferencial são amostrados de três diferentes grupos: um grupo contendo somente as partículas da população de fronteiras iguais ou superiores à da partícula, um grupo com as partículas da memória ou uma combinação dos dois grupos anteriores. Neste trabalho foram implementadas cinco estratégias evolutivas do DE para o operador de mutação diferencial que são: DE/Rand/1/Bin, DE/Rand/2/Bin, DE/Best/1/Bin, DE/Current-to-best/1/Bin e o DE/Current-to-rand/1/Bin. O diagrama do funcionamento do operador de mutação diferencial é apresentado pela [Figura 19](#) ao final desta seção.

Durante o desenvolvimento do MESH verificou-se que a mutação dos pesos seguindo a abordagem do C-DEEPSO e EPSO gerava frequentemente valores nos limites relativos a cada peso. Para reduzir esta frequência, a mutação dos pesos foi alterada para um valor amostrado da distribuição normal truncada, que é uma distribuição normal delimitada por um limite superior e inferior. Para os pesos dos termos de inercia, assimilação e cooperação o intervalo definido é $[0 - 1]$, para o peso referente ao crossover da mutação diferencial o intervalo é $[0 - 0,5]$ e para o peso do termo F empregado em algumas estratégias evolutivas do DE, o intervalo é $[0 - 1]$.

A seleção de quais partículas permanecerão na população para a próxima iteração no C-DEEPSO é feita comparando-se a partícula com seu clone após a aplicação da equação de movimento em ambas e escolhendo aquela que possuir a melhor solução. Já para o MESH, a seleção funciona de maneira igual ao do NSGA-II e esta é aplicada para toda a população original e suas cópias. Portanto, aplica-se o operador de ordenação por dominância rápida sobre todas partículas e seleciona-se as partículas pertencentes às primeiras fronteiras e no caso de uma fronteira ser maior que o tamanho restante da população, aplica-se o operador de distância de multidão.

Figura 17 – Diagrama do funcionamento do MESH.

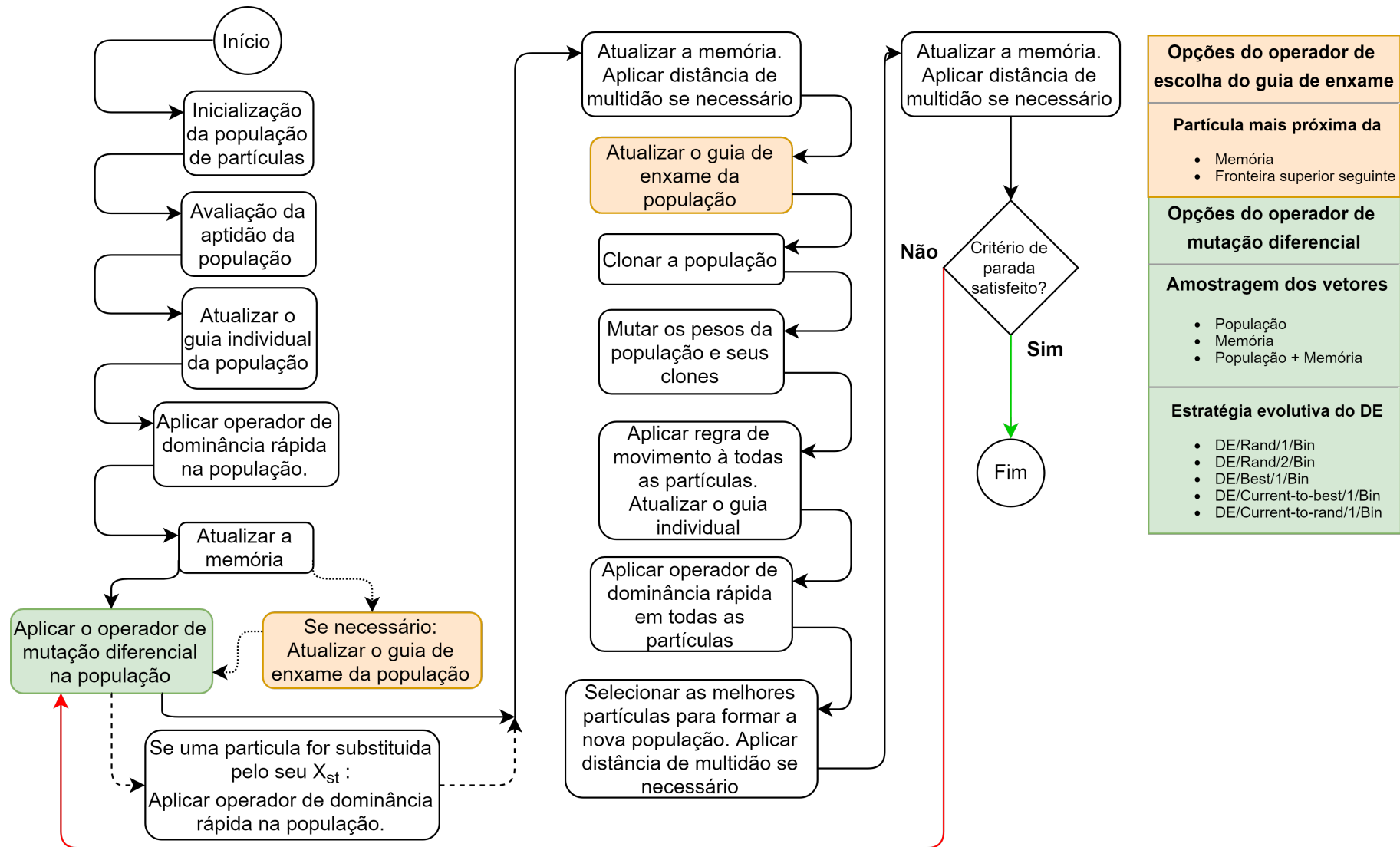


Figura 18 – Diagrama para a escolha do guia de enxame de uma partícula no MESH. No método apresentado à esquerda, todas partículas da população escolhem a partir da memória. No método apresentado à direita, a escolha é feita baseando-se na fronteira superior seguinte em relação à fronteira que a partícula pertence. As partículas da primeira fronteira por sua vez utilizam da memória.

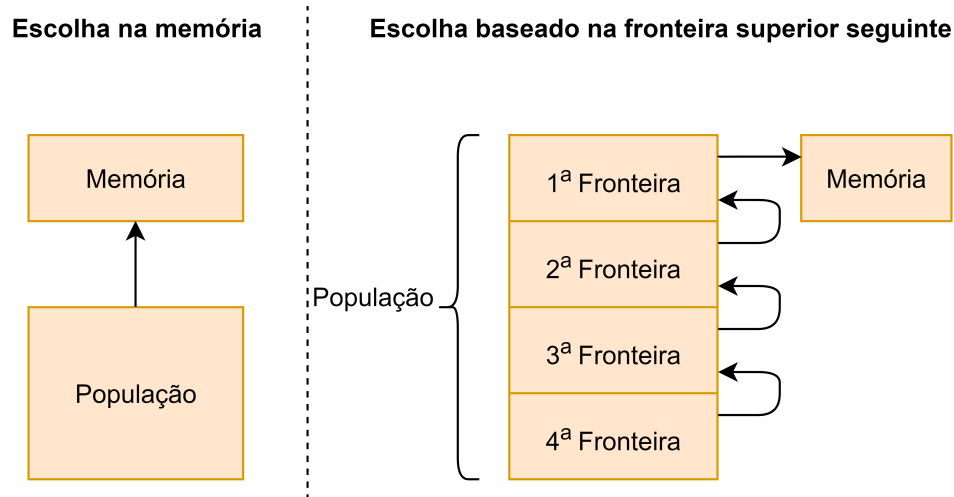
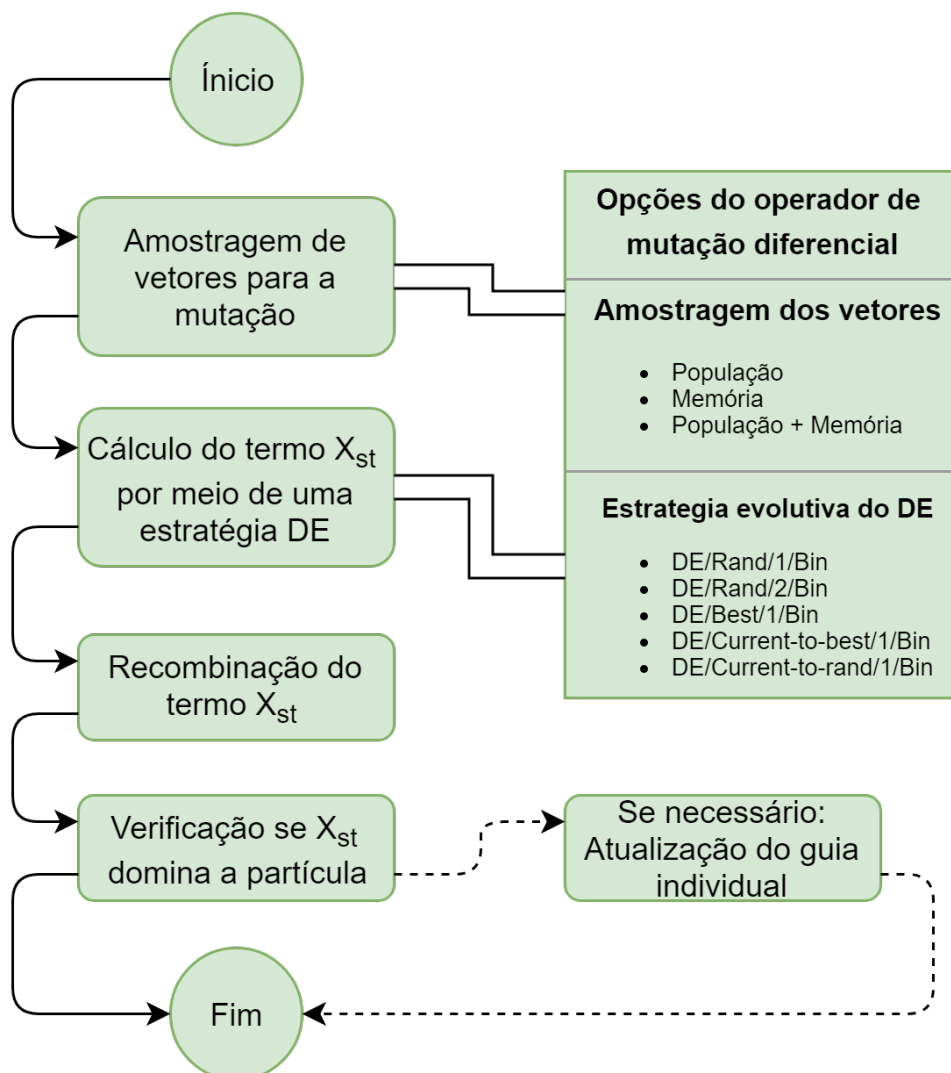


Figura 19 – Diagrama do funcionamento do operador de mutação diferencial do MESH.



4.2 Considerações finais

O nome do algoritmo se deu pelo fato de que o MESH incorpora a hibridização de diferentes algoritmos, no entanto a essência de sua busca se baseia nos paradigmas evolutivos e de enxame. Contudo o uso de diversas técnicas de outros algoritmos permitiu um número elevado de combinações para os parâmetros dos operadores do MESH. Isto gera uma inconveniência para o usuário na tarefa de decidir qual a melhor configuração do algoritmo para o seu uso.

Este trabalho não aborda a avaliação do custo computacional ou tempo de execução do MESH. Conforme visto no diagrama da [Figura 17](#), o MESH aplica o operador de dominância rápido do NSGA-II, que é o operador de maior custo do NSGA-II, por três vezes em cada iteração, além de gerar novas soluções com mecanismos baseado no DE e PSO. Logo, é esperado que o MESH tenha maior custo computacional do que os algoritmos MO não híbridos.

O MESH também realiza uma quantidade maior de avaliações na função de aptidão por iteração. Estas são feitas ao aplicar a regra de movimento nas partículas e no operador de mutação diferencial. Sendo que o operador de mutação diferencial pode não ser aplicado caso não haja a quantidade mínima de partículas para que estratégia evolutiva do DE seja empregada. Portanto, em uma população de N partículas, para cada iteração do MESH $2N$ avaliações serão feitas na movimentação das partículas e de 0 à N avaliações serão feitas no operador de mutação diferencial.

5 Proposta de modelo matemático para a geração de energia em UHE em cascata

O modelo de UHE em cascata proposto neste trabalho é fundamentado no modelo apresentado na [Subseção 2.3.2](#). Este modelo tem como parâmetros de entrada a demanda de energia a ser produzida pela UHE, a altura da queda bruta e os coeficientes operativos da unidade geradora. As variáveis de decisão são as vazões turbinadas para cada turbina. Além disto o modelo é estático, pois dados os parâmetros de entrada não há uma mudança no sistema ao longo do tempo. Por ser estático, o balanço hídrico é desconsiderado e como a altura da queda bruta é um parâmetro de entrada, o volume do reservatório também não é considerado no modelo.

Ao adicionar o balanço hídrico, temos um sistema dinâmico, uma vez que o nível do reservatório é alterado ao longo do tempo. Para um cenário de UHEs em cascata o balanço hídrico é essencial para o modelo, pois há uma interferência no nível do reservatório de uma UHE por influência das vazões defluentes de outra UHE. Para uma UHE que está em cascata com outra, o volume de seu reservatório em um tempo t pode ser descrito pela [Equação \(31\)](#),

$$V_{u,t} = V_{u,t-1} + Qa_{u,t} + Qt_{w,td} + Qv_{w,td} - Qt_{u,t-1} - Qv_{u,t-1} - E_{u,t-1} \cdot A_{u,t-1} \quad (31)$$

na qual,

- u e w são índices que identificam uma usina, neste caso $u \neq w$
- td é tempo necessário para o deslocamento da água entre as usinas u e w
- V é o volume do reservatório;
- Qa é a vazão afluente;
- Qt é a vazão turbinada;
- Qv é a vazão vertida;
- E é a evaporação líquida;
- A é a área ocupada pela água no reservatório.

A unidade da variável td deve ser a mesma da variável t , as demais unidades seguem as definições feitas na [Subseção 2.3.1](#): hm^3 para o volume, $m^3.s^{-1}$ para as vazões, mm por dia para a evaporação e km^2 para a área.

Uma vez que a evolução do nível do reservatório é considerada no modelo, o valor

da queda bruta deixa de ser um parâmetro de entrada e se torna uma variável depende das cotas jusante e montante, conforme definido na [Subseção 2.3.1](#), i.e., $hb = f_{cm} - f_{cj}$.

Com as considerações anteriores, o modelo proposto neste trabalho é formulado por:

$$\text{Maximize } F_1(Qt_{uj,t} \dots Qt_{UJ_u,t}) = \frac{1}{U} \sum_{u=1}^U \left(\frac{\sum_{j=1}^{J_u} ph_{uj,t}}{\sum_{j=1}^{J_u} Qt_{uj,t}} \right), \quad (32)$$

$$\text{Maximize } F_2(Qt_{uj,t} \dots Qt_{UJ_u,t}) = \frac{1}{U} \sum_{u=1}^U \frac{V_{u,t}}{V_u^{max}}. \quad (33)$$

Sujeito às seguintes restrições:

$$[1.] \quad V_{u,t} = V_{u,t-1} + c \left(Qa_{u,t} + Qt_{w,t} + Qv_{w,t} - \sum_{j=1}^{J_u} Qt_{uj,t-1} - Qv_{u,t-1} \right) - E_{u,t-1} \cdot A_{u,t-1},$$

$$[2.] \quad Dm_{u,t}(1 - \varepsilon) \leq \sum_{j=1}^{J_u} ph_{uj,t} \leq Dm_{u,t}(1 + \varepsilon),$$

$$[3.] \quad V_u^{min} \leq V_{u,t} \leq V_u^{max},$$

$$[4.] \quad Qd_u^{min} \leq Qv_{u,t} + \sum_{j=1}^{J_u} Qt_{uj,t} \leq Qd_u^{max},$$

$$[5.] \quad Qt_{uj}^{min} \leq Qt_{uj,t} \leq Qt_{uj}^{max},$$

$$[6.] \quad Qv_{u,t} \leq Qv_u^{max},$$

$$[7.] \quad V_{u,t} > V_u^{max} \implies Qv_{u,t} \geq \frac{1}{c} (V_{u,t} - V_u^{max}),$$

$$[8.] \quad ph_{uj}^{min} \cdot Z_{uj,t} \leq ph_{uj,t} \leq ph_{uj}^{max} \cdot Z_{uj,t},$$

$$[9.] \quad Z_{uj,t} \in \{0,1\}.$$

Sendo:

$$ph_{uj,t} = gk \cdot [\rho_{0uj} + \rho_{1uj}hl_{uj,t} + \rho_{2uj}Qt_{uj,t} + \rho_{3uj}hl_{uj,t}Qt_{uj,t} + \rho_{4uj}hl_{uj,t}^2 + \rho_{5uj}Qt_{uj,t}^2] \\ \cdot [Hb_{u,t} - \Delta_{Huj,t}] \cdot Qt_{uj,t} ,$$

$$hl_{uj,t} = Hb_{u,t} - \Delta_{Huj,t} ,$$

$$Hb_{u,t} = fcm_{u,t} - fcj_{u,t} ,$$

$$fcm_{u,t} = a_{0,u} + a_{1,u} \cdot V_{u,t} + a_{2,u} \cdot V_{u,t}^2 + a_{3,u} \cdot V_{u,t}^3 + a_{4,u} \cdot V_{u,t}^4 ,$$

$$fcj_{u,t} = b_{0,u} + b_{1,u} \cdot \left(\sum_{j=1}^{J_u} Qt_{u,t} + Qv_{u,t} \right) + b_{2,u} \cdot \left(\sum_{j=1}^{J_u} Qt_{u,t} + Qv_{u,t} \right)^2 \\ + b_{3,u} \cdot \left(\sum_{j=1}^{J_u} Qt_{u,t} + Qv_{u,t} \right)^3 + b_{4,u} \cdot \left(\sum_{j=1}^{J_u} Qt_{u,t} + Qv_{u,t} \right)^4 .$$

Na qual:

- u é o índice que identifica uma UHE;
- U é número total de UHE no sistema;
- j é uma unidade geradora em uma UHE;
- J_u é número total de unidades geradoras na UHE u ;
- t é o índice associado ao tempo;
- $ph_{uj,t}$ é a potência gerada pela unidade geradora j da UHE u no tempo t em MW ;
- $V_{u,t}$ é o volume do reservatório da usina u no tempo t em hm^3 ;
- c é uma constante de conversão de uma vazão em $m^3 \cdot s^{-1}$ em um volume em hm^3 , para um período de tempo igual ao tempo t ;
- $Qa_{u,t-1}$ é a vazão afluente que chega ao reservatório da UHE u no tempo $t - 1$ em $m^3 \cdot s^{-1}$;
- w é o índice de uma UHE em que sua vazão defluente chega no reservatório da UHE u ;
- td é o tempo necessário para que a água se desloque da UHE w até a UHE u ;
- $Qt_{w,td}$ é a vazão turbinada em $m^3 \cdot s^{-1}$ que chega ao reservatório da UHE u no tempo td oriunda da UHE w ;
- $Qv_{w,td}$ é a vazão vertida em $m^3 \cdot s^{-1}$ que chega ao reservatório da UHE u no tempo td oriunda da UHE w ;

- $Qt_{uj,t-1}$ é a vazão turbinada em $m^3 \cdot s^{-1}$ que passa pela unidade geradora j da UHE u no tempo $t - 1$;
- $Qv_{u,t-1}$ é a vazão vertida em $m^3 \cdot s^{-1}$ que passa pelo vertedouro da UHE u no tempo $t - 1$;
- $E_{u,t-1}$ é a evaporação líquida no tempo $t - 1$ em mm por dia;
- $A_{u,t-1}$ é a área ocupada pela água no reservatório da UHE u no tempo $t - 1$ em km^2 ;
- $Dm_{u,t}$ é a demanda de energia requerida à UHE u no tempo t em MW ;
- ε é a variação de erro tolerada na potência produzida pela UHE;
- V_u^{min} e V_u^{max} são respectivamente os volumes nas cotas mínima e máxima do reservatório da UHE u ;
- Qd_u^{min} e Qd_u^{max} são os limites para a vazão defluente da UHE u ;
- Qt_{uj}^{min} e Qt_{uj}^{max} são os limites para a vazão turbinada na unidade geradora j da UHE u ;
- Qv_u^{max} é limite máximo para a vazão vertida da UHE u ;
- $Z_{uj,t}$ indica o estado de operação da unidade geradora j na UHE u , sendo 0 para desativada e 1 para ativa;
- g é a aceleração da gravidade em $9,8m \cdot s^{-2}$;
- k é a constante $10^{-3}kg \cdot m^{-1}$ que converte cavalo vapor, HP , em MW ;
- $\rho_{0uj} \dots \rho_{5uj}$ são os coeficientes operativos da unidade geradora j da UHE u ;
- $hl_{uj,l}$ é altura da queda líquida na unidade geradora j na UHE u no tempo t ;
- $\Delta_{Huj,t}$ são as perdas hidráulicas que ocorrem nos condutos que levam a água até a unidade geradora j na UHE u no tempo t ;
- $Hb_{u,t}$ é a altura da queda bruta na UHE u no tempo t ;
- $fcm_{u,t}$ é a altura da cota montante da UHE u no tempo t ;
- $a_{0,u} \dots a_{4,u}$ são os coeficientes para o polinômio de quarta ordem da UHE u que definem $fcm_{u,t}$;
- $fcj_{u,t}$ é a altura da cota jusante da UHE u no tempo t ;
- $b_{0,u} \dots b_{4,u}$ são os coeficientes para o polinômio de quarta ordem da UHE u que definem $fcj_{u,t}$.

O primeiro objetivo do modelo é maximizar a eficiência média do sistema. A fração com os somatórios de $ph_{uj,t}$ e $Qt_{uj,t}$, da [Equação \(32\)](#), determina a quantidade de energia que a usina u é capaz de produzir com um determinado volume de água. Maximizar esta função implica em gerar energia com uma vazão de água menor. O segundo objetivo do modelo é inspirado do trabalho de [Lisboa, Guedes e Vieira \(2013\)](#) e consiste em maximizar a média dos níveis de água nos reservatórios do sistema. Manter um alto nível nos reservatórios do sistema aumenta a robustez do sistema para períodos de seca futuros. Ao mesmo tempo, quanto maior o nível do reservatório, maior sua cota montante e a altura da queda bruta, consequentemente maior a eficiência da UHE.

A primeira restrição se refere ao balanço hídrico do reservatório de uma UHE do

sistema. É esta equação que modela o acoplamento da operação dos reservatórios das UHE no sistema. A segunda restrição indica que cada usina do sistema deve entregar uma potência aproximadamente igual à demanda solicitada. O ONS aceita um erro ε de até 0,5% acima ou abaixo do valor solicitado. A terceira restrição limita o volume do reservatório a um intervalo relativo aos limites das cotas de mínima e máxima de operação. A quarta restrição indica que a vazão defluente de uma usina deve respeitar um intervalo limite. Estes limites servem de controle para evitar cheias e inundações em regiões no rio jusante à UHE e também para o uso da água para navegações e pelo ecossistema no rio e em suas proximidades. A quinta restrição informa que as vazões turbinadas devem respeitar os limites de capacidade de suas respectivas unidades geradoras. A sexta restrição impõe um limite máximo para a vazão vertida. Apesar dos vertedouros serem projetados suportar uma vazão afluyente decamilenar, i.e., a maior vazão que ocorreria, estatisticamente, a cada 10 mil anos, algumas UHE limitam a vazão vertida por fatores operacionais e pelo os mesmos motivos apresentados para limitar a vazão defluente. A sétima restrição informa que caso o volume do reservatório supere o seu limite máximo operativo, o excesso de água deve ser eliminado pelo vertedouro. A oitava restrição indica que a potência gerada também deve respeitar os limites da capacidade de sua unidade geradora. A nona restrição indica que as unidades geradoras possuem apenas uma zona de operação.

As variáveis estáticas de entrada do modelo são: as vazões afluentes à montante de cada UHE em $m^3 \cdot s^{-1}$, a demanda de energia solicitada para cada UHE em MW e os coeficientes operativos das unidades geradoras de cada UHE. As variáveis dinâmicas são: os volumes dos reservatórios de cada UHE em hm^3 e as vazões defluentes de cada UHE em $m^3 \cdot s^{-1}$ que estão à montante do reservatório de uma outra UHE.

6 Experimentos e análise dos resultados

6.1 Ambiente e parâmetros dos experimentos

Os experimentos realizados no presente trabalho utilizam três algoritmos evolucionários: o MESH desenvolvido neste trabalho, o NSGA-II e o SPEA2. Todos os algoritmos foram implementados com a linguagem de programação Python em sua versão 3.5.4. Para os cálculos de operações matemáticas trigonométricas e exponenciais, e também para a geração da maioria dos números aleatórios, utilizou-se da biblioteca Numpy em sua versão 1.15. A biblioteca Scipy na versão 1.1.0 foi aplicada na geração de números aleatórios com a distribuição normal truncada. Toda a análise estatística definida na metodologia foi executada com o software Matlab na versão R2017a.

Em todas execuções dos três algoritmos, o critério de parada foi o limite de 15000 avaliações da função de aptidão. O tamanho definido para todos os tipos de populações em todos algoritmos é de 50. Este tamanho é válido tanto para a população principal, quanto para as populações secundárias como a memória, de cópia, arquivo ou de filhos. No NSGA-II, a probabilidade de mutação é de 2% e o cruzamento é feito de forma a se atingir uma população de filhos com tamanho de 50 indivíduos. No SPEA2, a taxa de cruzamento é de 0.7.

6.2 Experimento 1: validação das configurações do MESH

O primeiro experimento deste trabalho visa avaliar quais as melhores configurações de operadores do MESH apresentam melhor desempenho e destas escolher duas para serem usadas nos experimentos seguintes. Para simplificar a leitura das comparações feitas neste experimento, adotou-se a nomenclatura Ex/Vy/Dz, na qual x, y, z indicam o índice da opção escolhida para o operador de escolha do guia de enxame, o grupo de vetores e a estratégia DE do operador de mutação diferencial respectivamente. Os índices de cada opção são definidos nos parágrafos seguintes.

O operador de escolha do guia de enxame permite duas opções:

- (E1) Selecionar a partícula mais próxima na memória.
- (E2) Selecionar a partícula mais próxima na fronteira superior seguinte à fronteira da partícula atual.

Para o operador de mutação diferencial, o usuário deve primeiro definir qual o

grupo de partículas para a amostragem dos vetores. Quanto aos grupos para a amostragem dos vetores temos três opções:

- (V1) População.
- (V2) Memória.
- (V3) Combinação da População e da memória.

Por fim o usuário deve definir qual a estratégia evolutiva DE será aplicada nos vetores. Neste trabalho foram implementadas cinco estratégias evolutivas do DE:

- (D1) DE/Rand/1/Bin.
- (D2) DE/Rand/2/Bin.
- (D3) DE/Best/1/Bin.
- (D4) DE/Current-to-best/1/Bin.
- (D5) DE/Current-to-rand/1/Bin.

Como exemplo, E1/V3/D3 é a configuração em que o guia de enxame é escolhido pela memória e a mutação diferencial é feita com vetores amostrados da população e memória sobre a estratégia DE/Best/1/Bin.

Devido ao número de diferentes configurações neste experimento, apenas as funções ZDT3, ZDT4 e ZDT6 foram utilizadas. Para facilitar a leitura, adotou-se uma relação dos índices dos gráficos para as configurações do MESH. A [Tabela 2](#) mostra esta relação.

Tabela 2 – Relação índice dos gráficos com as configurações de MESH.

Índice	Configuração	Índice	Configuração	Índice	Configuração
1	E1/V1/D1	11	E1/V3/D1	21	E2/V2/D1
2	E1/V1/D2	12	E1/V3/D2	22	E2/V2/D2
3	E1/V1/D3	13	E1/V3/D3	23	E2/V2/D3
4	E1/V1/D4	14	E1/V3/D4	24	E2/V2/D4
5	E1/V1/D5	15	E1/V3/D5	25	E2/V2/D5
6	E1/V2/D1	16	E2/V1/D1	26	E2/V3/D1
7	E1/V2/D2	17	E2/V1/D2	27	E2/V3/D2
8	E1/V2/D3	18	E2/V1/D3	28	E2/V3/D3
9	E1/V2/D4	19	E2/V1/D4	29	E2/V3/D4
10	E1/V2/D5	20	E2/V1/D5	30	E2/V3/D5

Para comparar as configurações nas funções *benchmark*, segue-se a sequência de testes estatísticos proposta na metodologia do [Capítulo 3](#). Esta sequência de testes estatísticos é aplicada para cada uma das métricas de comparação. Entretanto neste primeiro experimento em vez de uma análise independente das duas métricas, segue-se uma sequência entre elas.

Primeiramente avalia-se os hipervolumes das configurações até se obter as duas melhores configurações ou então um grupo de configurações estatisticamente equivalentes

sobre o critério da métrica de hipervolume. Com este grupo reduzido, avalia-se então a distância geracional com objetivo de se restringir mais este grupo para se encontrar as duas melhores configurações. Se após as avaliações com a distância geracional, houver um conjunto com mais de duas configurações, será utilizado a métrica do número de avaliações de aptidão. Como a análise dos resultados dos experimentos é extensa, uma síntese é apresentada pela [Tabela 3](#), nesta tabela os índices das configurações seguem a relação definida pela [Tabela 2](#).

Tabela 3 – Síntese das análises dos resultados do experimento 1.

Métrica avaliada	Índices das configurações escolhidas em cada etapa
ZDT6	
Hipervolume	1 2 6 8 9 11 12 14 15 17 21 22 26 27
Distância geracional	1 2 6 9 11 12 14 15 17 21 22 26 27
Número de avaliações	1 9 14 15
ZDT4	
Hipervolume	1 2 12 20 26
2ª avaliação	1 20
ZDT3	
Hipervolume	6 12 15 17 21 27
Distância geracional	17 21

Pela [Tabela 3](#), nota-se que a função ZDT6 é a que possui a maior quantidade de configurações com desempenho semelhante, havendo quase a metade delas ao fim das avaliações das duas primeiras métricas. Foi necessário avaliar todas as três métricas propostas e por fim não foi possível atingir o objetivo de duas configurações, sendo quatro configurações escolhidas como resultado final. Já a função ZDT4 é a mais simples de se escolher as duas configurações, sendo necessário apenas a avaliação do hipervolume. A ZDT3 semelhante à ZDT4 possui poucas configurações com desempenho semelhante e após a avaliação da distância geracional obtêm-se as duas melhores configurações.

A [Tabela 4](#) mostra a compilação das configurações escolhidas pelos resultados dos experimentos na ZDT6, ZDT4 e ZDT3. Entre as oito configurações, apenas a E1/V1/D1 está presente em mais de uma função ZDT. Todas as configurações para ZDT6 utilizam o operador de escolha do guia de exame baseado na memória, enquanto para a ZDT3 e ZDT4 as duas opções estão representadas. A estratégia evolutiva DE/Best/1/Bin (D3) não está representada por nenhuma das três ZDTs, já a DE/Current-to-best/1/Bin (D4) é utilizada apenas na ZDT6. Considerando que a configuração E1/V1/D1 foi escolhida na ZDT6 e ZDT4, a estratégia DE/Rand/1/Bin (D1) é a mais frequente nas três ZDTs.

A estratégia de amostragem dos vetores combinando-se a população e a memória (V3) está presente apenas nas escolhas da ZDT6. A amostragem utilizando apenas a

população (V1) é a mais frequente se considerarmos duas vezes sua escolha em E1/V1/D1. A primeira configuração escolhida para ser usada nos outros experimentos é a E1/V1/D1, por ter sido a única representada em mais de uma função ZDT. Com o objetivo de ter uma maior diversidade de operadores, a segunda configuração escolhida é a E2/V2/D1 utilizada na ZDT3.

Tabela 4 – Compilação das melhores configurações do MESH para a ZDT6, ZDT4 e ZDT3.

Configuração	ZDT6	ZDT4	ZDT3
E1/V1/D1	✓	✓	
E1/V2/D2	✓		
E1/V2/D4	✓		
E1/V3/D4	✓		
E1/V3/D5	✓		
E2/V1/D2			✓
E2/V1/D5		✓	
E2/V2/D1			✓

6.3 Experimento 2: avaliação de desempenho em funções *benchmark*

O segundo experimento compara o MESH com os algoritmos MO NSGA-II e SPEA2 avaliando-os nas cinco funções *benchmark* ZDT. As duas configurações escolhidas no experimento 1 são utilizadas nas funções ZDT1 e ZDT2. Enquanto para as funções ZDT3, ZDT4 e ZDT6, apenas a configuração escolhida na respectiva ZDT é aplicada. O objetivo do experimento é verificar se o desempenho do MESH se mostra competitivo com os algoritmos MO clássicos.

6.3.1 Paretos combinados

A [Figura 20](#) apresenta os paretos combinados dos algoritmos para as cinco funções ZDT. O MESH gera paretos sobre a curva do pareto ótimo em quatro funções. Na ZDT2, a maior parte das execuções do MESH em ambas configurações, geram paretos que possuem diferentes soluções com a mesma posição no espaço objetivo. Este comportamento, entretanto, não é visível pela curva do pareto ótimo, sua causa é explicada na análise dos hipervolumes.

Dentre os algoritmos clássicos, o NSGA-II apresenta paretos superiores aos do SPEA2 exceto na ZDT2, sendo que na ZDT3 e ZDT6 seus paretos estão sobre a curva do pareto ótimo. O SPEA2 apesar de exibir os piores resultados na maioria das funções, seu pareto combinado na ZDT6 possui a maioria das soluções sobre a curva do pareto ótimo e nas ZDT2 e ZDT4 seu pareto é próximo do pareto ótimo. De forma geral, os gráficos dos paretos combinados sugerem que o MESH possui um desempenho melhor do que os outros dois algoritmos.

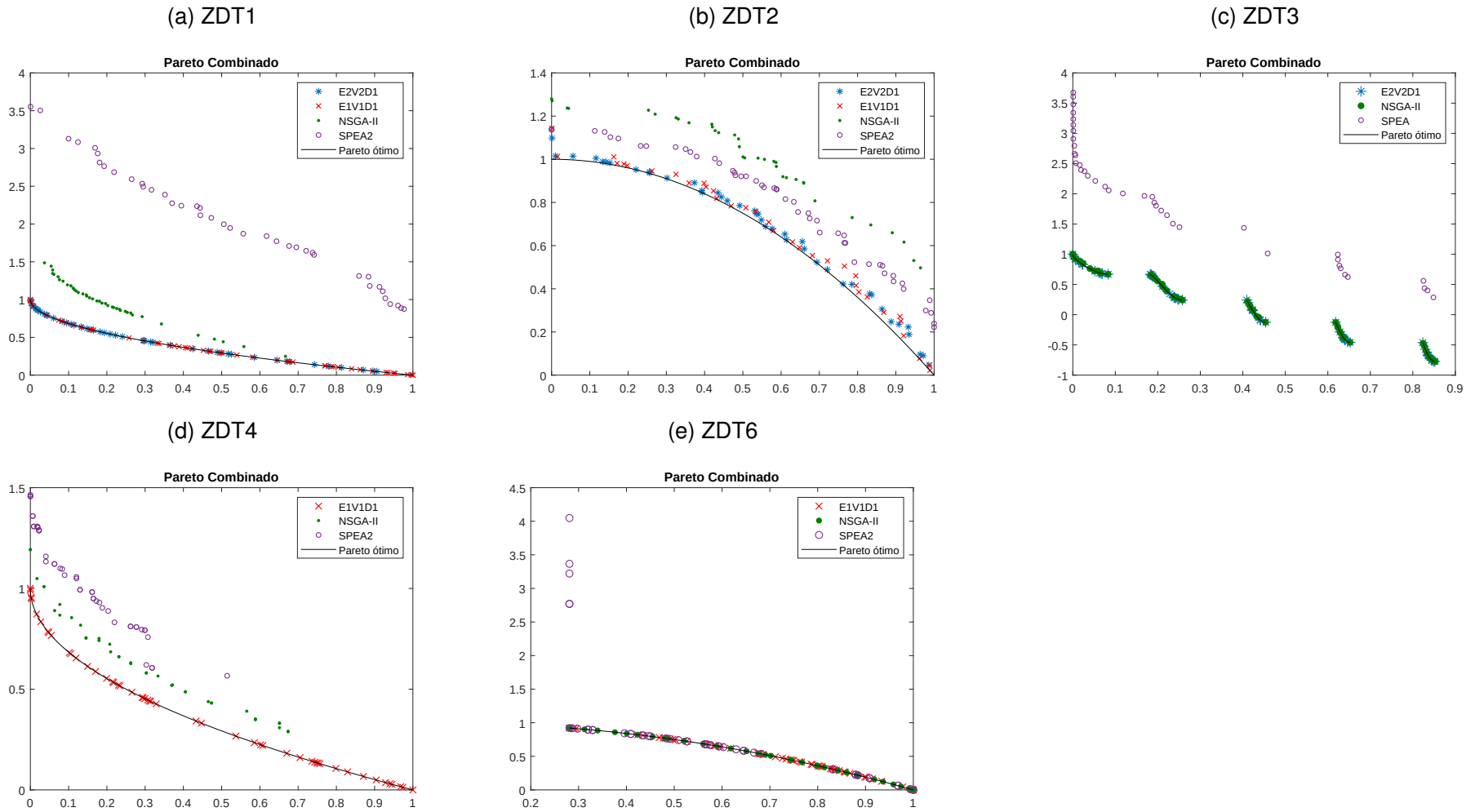
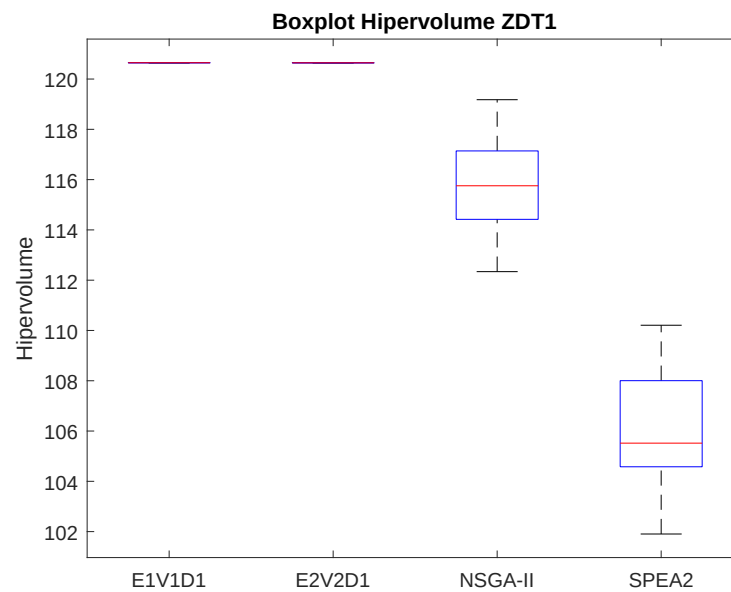


Figura 20 – Pareto combinado dos algoritmos nas funções ZDT.

6.3.2 Análise das métricas de comparação

A Figura 21 apresenta o *boxplot* dos hipervolumes para os três algoritmos. Os resultados do MESH apresentam menor variabilidade do que dos outros dois algoritmos, por isso suas caixas aparecem como linhas. Nenhum dos algoritmos possuem interseção entre suas caixas, com isto afirma-se que o MESH apresentou um desempenho superior aos NSGA-II e SPEA2. A ANOVA dos hipervolumes tem o p-valor de $2,751e^{-25}$, confirmando a existência de uma diferença significativa entre os algoritmos. O teste de Tukey da Figura 22 confirma a análise do gráfico *boxplot* anterior.

Figura 21 – *Boxplot* dos hipervolumes dos três algoritmos na função ZDT1.

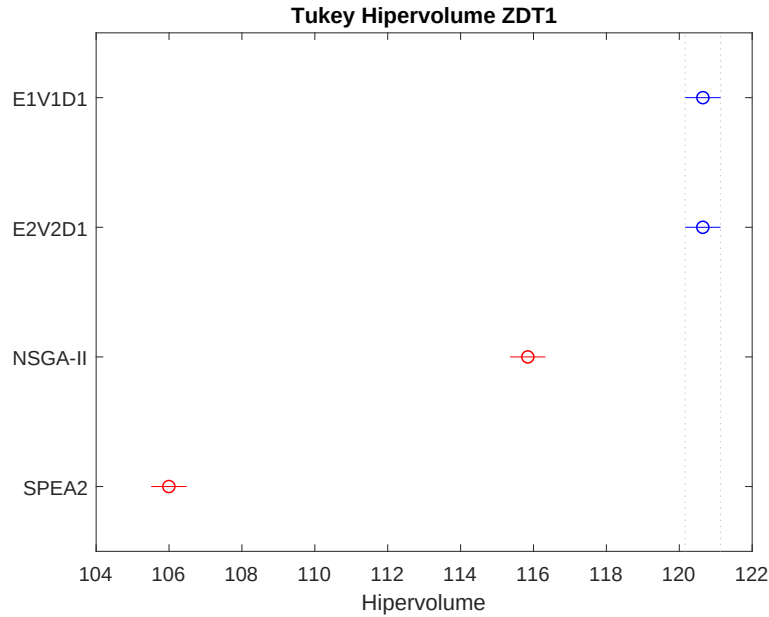


As demais funções também são avaliadas como feito na ZDT1, os seus resultados são apresentados de forma resumida na Tabela 5, contendo os maiores e menores hipervolumes encontrados em cada função ZDT e a classificação dos três algoritmos de acordo com o teste de Tukey. O MESH gera os paretos com os maiores hipervolumes em todas funções ZDT, enquanto o SPEA2 gera os menores hipervolumes. Pelo teste de Tukey, o MESH gera os melhores hipervolumes nas funções ZDT1 e ZDT4. Nas funções ZDT3 e ZDT6 o MESH e o NSGA-II são equivalentes. Já o SPEA2 é o algoritmo que apresenta os piores hipervolumes.

Tabela 5 – Análise dos hipervolumes dos algoritmos nas funções ZDT.

ZDT	Hipervolume			ANOVA P-valor	Classificação Tukey		
	Ótimo	Maior Encontrado	Menor Encontrado		1º	2º	3º
1	120,657446	E1V1D1:120,6526	SPEA2:102,8956	$2,751e^{-25}$	E1V1D1 E2V2D1	NSGA-II	SPEA2
2	120,324398	E1V1D1:119,9622	SPEA2:109,8298	$1,789e^{-12}$	E1V1D1 E2V2D1	NSGA-II	SPEA2
3	128,773637	E2V2D1:128,5779	SPEA2:98,2654	$2,942e^{-43}$	E2V2D1 NSGA-II	SPEA2	—
4	120,657446	E1V1D1:120,6532	SPEA2:108,3511	$1,225e^{-13}$	E1V1D1	NSGA-II	SPEA2
6	117,511658	E1V1D1:117,5051	SPEA2:112,933	$1,117e^{-8}$	E1V1D1 NSGA-II	SPEA2	—

Figura 22 – Teste de Tukey dos hipervolumes dos três algoritmos na função ZDT1.



Durante o cálculo dos hipervolumes dos paretos gerados pelo MESH na ZDT2, notou-se que um conjunto de paretos nas duas configurações possuíam o mesmo hipervolume de valor 110. Nesta função, o pareto ótimo é formado se $x_2 = 1 - x_1^2$ e $x_3 \dots x_n = 0$, se o vetor x for nulo, os valores da função ZDT2 será $F = (0 \ 1)$. O valor do hipervolume gerado com o ponto $F = (0 \ 1)$ e o ponto de referência $(11 \ 11)$ é de 110.

Por questões de arredondamento em operações com números reais, é possível que um vetor x com $x_1 = x_2 = 0$ e com valores muito próximos de 0 para $x_3 \dots x_n$ (como por exemplo valores na ordem de 10^{-17} a 10^{-21}), tenha o mesmo resultado na ZDT2 que o vetor x completamente nulo. A situação descrita anteriormente, permite que soluções diferentes no espaço de busca tenham o mesmo valor no espaço objetivo.

Nas execuções em que o pareto final do MESH possui um hipervolume de 110, o algoritmo encontra em um determinado momento uma solução muito próxima do vetor nulo e nas iterações seguintes novas soluções semelhantes são geradas. Na maioria das execuções quando esta situação ocorre o algoritmo fica preso a um pareto formado por soluções com o mesmo valor de $(0 \ 1)$ na função ZDT2.

Apesar da análise estatística feita validar a vantagem do MESH sobre os outros algoritmos, não é sensato afirmar que o algoritmo é capaz para solucionar a ZDT2, quando a maioria de suas execuções ficou presa próxima de um ponto do pareto ótimo. A análise para as distâncias geracionais é apresentada pela [Tabela 6](#), na qual temos resultados semelhantes aos da análise com os hipervolumes. O MESH obtem as menores distâncias e o por sua vez SPEA2 as maiores. Nas funções ZDT3 e ZDT6, o NSGA-II e MESH são equivalentes. Nas demais funções o MESH está mais próximo do pareto ótimo do que os

outros dois algoritmos.

Tabela 6 – Análise das distâncias geracionais dos algoritmos nas funções ZDT.

ZDT	Menor Distância	Maior Distância	P-valor ANOVA	Classificação Tukey		
				1º	2º	3º
1	E1V1D1:0,00112	SPEA2:0,11942	$1,041e^{-5}$	E1V1D1 E2V2D1	NSGA-II	SPEA2
2	E2V2D1:0,001013	SPEA2:0,13825	$2,8324e^{-6}$	E1V1D1 E2V2D1	NSGA-II	SPEA2
3	E2V2D1:0,001062	SPEA2:0,11251	$1,1214e^{-5}$	E2V2D1 NSGA-II	SPEA2	–
4	E1V1D1:0,001426	SPEA2:0,15695	$1,8512e^{-6}$	E1V1D1	NSGA-II	SPEA2
6	E1V1D1:0,001323	SPEA2:0,00956	$2,365e^{-4}$	E1V1D1 NSGA-II	SPEA2	–

Para diferenciar o MESH e o NSGA-II nas funções ZDT3 e ZDT6, foi analisado a métrica do número de avaliações na função de aptidão. Na ZDT3, o MESH faz em média 10433 avaliações e o NSGA-II 13580 avaliações. Pela ANOVA com o p-valor de $2,587e^{-19}$ e o teste de Tukey, confirma-se que o MESH é mais rápido para atingir um pareto próximo do ótimo. Na ZDT6, o MESH faz em média 9854 avaliações e o NSGA-II 11750 avaliações. Pela ANOVA com o p-valor de $3,982e^{-11}$ e o teste de Tukey, também confirma que o MESH é mais rápido que o NSGA-II.

6.3.3 Considerações finais

O objetivo do experimento 2 é comparar o desempenho do MESH com os algoritmos MO clássicos NSGA-II e SPEA2. Para cumprir este objetivo foram feitos experimentos com os três algoritmos nas funções ZDTs. A comparação foi feita utilizando as métricas: hipervolume, distância geracional e número de avaliações na função de aptidão como desempate.

A Tabela 7 apresenta a média da diferença entre os hipervolumes encontrados e o hipervolume máximo para cada uma das funções ZDTs. De modo geral, o MESH tem o melhor desempenho, tendo encontrado soluções com uma diferença na ordem de 10^{-2} do hipervolume máximo em quatro das cinco funções.

Tabela 7 – Média da diferença entre os hipervolumes encontrados e o hipervolume máximo no experimento 2.

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
E1/V1/D1	0,0099 $\pm 0,0035$	6,0437 $\pm 0,4676$	–	0,0399 $\pm 0,0289$	0,0161 $\pm 0,0194$
E2/V2/D1	0,0118 $\pm 0,0036$	6,3176 $\pm 0,4692$	0,0179 $\pm 0,0378$	–	–
NSGA-II	2,4893 $\pm 0,5647$	8,6578 $\pm 0,1964$	0,0184 $\pm 0,0372$	5,1120 $\pm 1,5469$	0,0188 $\pm 0,0203$
SPEA2	7,5684 $\pm 2,5878$	9,8702 $\pm 0,8971$	10,4569 $\pm 2,0950$	11,7896 $\pm 2,7735$	3,0075 $\pm 0,4491$

A média das distâncias geracionais dos algoritmos em cada uma das funções ZDTs é apresentada na [Tabela 8](#). Do mesmo modo que na análise com os hipervolumes, o MESH possui as menores distâncias geracionais com valores na ordem de 10^{-3} em todas ZDTs exceto na ZDT2 com a configuração E2/V2/D1. É importante ressaltar que os resultados da ZDT2 consideram apenas as execuções que foram capazes de gerar um pareto completo.

Tabela 8 – Média das distancias geracionais do experimento 2.

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
E1/V1/D1	0,0013 $\pm 7,758e^{-5}$	0,0082 $\pm 7,6982e^{-5}$	—	0,0023 $\pm 7,2063e^{-4}$	0,0025 $\pm 7,4569e^{-4}$
E2/V2/D1	0,0013 $\pm 8,7441e^{-5}$	0,0102 $\pm 7,5698e^{-5}$	0,0023 $\pm 5,3378e^{-5}$	—	—
NSGA-II	0,0062 $\pm 3,3248e^{-4}$	0,0229 $\pm 3,2354e^{-5}$	0,0025 $\pm 6,0365e^{-5}$	0,0058 $\pm 8,0057e^{-4}$	0,0021 $\pm 8,4439e^{-4}$
SPEA2	0,0575 $\pm 6,2237e^{-5}$	0,0179 $\pm 7,6699e^{-5}$	0,0311 $\pm 2,8457e^{-5}$	0,0274 $\pm 1,3365e^{-5}$	0,0065 $\pm 5,1123e^{-4}$

Neste experimento, o MESH se mostrou um algoritmo competitivo sendo capaz de encontrar soluções próximas às ótimas em quatro das cinco funções. Quando comparado aos algoritmos NSGA-II e SPEA2, o MESH gera soluções superiores em duas funções. E nas duas funções em que é equivalente ao NSGA-II, o MESH é mais rápido para atingir um pareto próximo ao ótimo.

6.4 Experimento 3: avaliação em problemas reais e validação do modelo de UHE em cascata proposto

O terceiro experimento deste trabalho tem dois objetivos: avaliar o desempenho do MESH em problemas reais e validar o modelo de geração de energia de UHEs em cascata proposto. Assim como no experimento 2, o MESH será comparado com o NSGA-II e o SPEA2 na otimização de dois modelos multiobjetivo de geração de energia hidrelétrica: o modelo de [Marcelino et al. \(2015\)](#) apresentado ao final da [Subseção 2.3.2](#) e o modelo de UHEs em cascata proposto neste trabalho. Como a aplicação de todas as configurações do MESH tem custo computacional elevado, apenas as configurações E1/V1/D1 e E2/V2/D1 foram utilizadas, dado que estas foram as escolhidas no experimento 1.

Na validação do modelo proposto, pretende-se avaliar se o seu uso é vantajoso, para isto será feito uma simulação aplicando o modo controle conjunto. Os resultados desta simulação serão comparados com os resultados otimizados encontrados pelos algoritmos, caso o controle otimizado seja superior, valida-se os objetivos do modelo proposto.

6.4.1 Avaliação para o modelo de distância entre os controles otimizado e conjunto

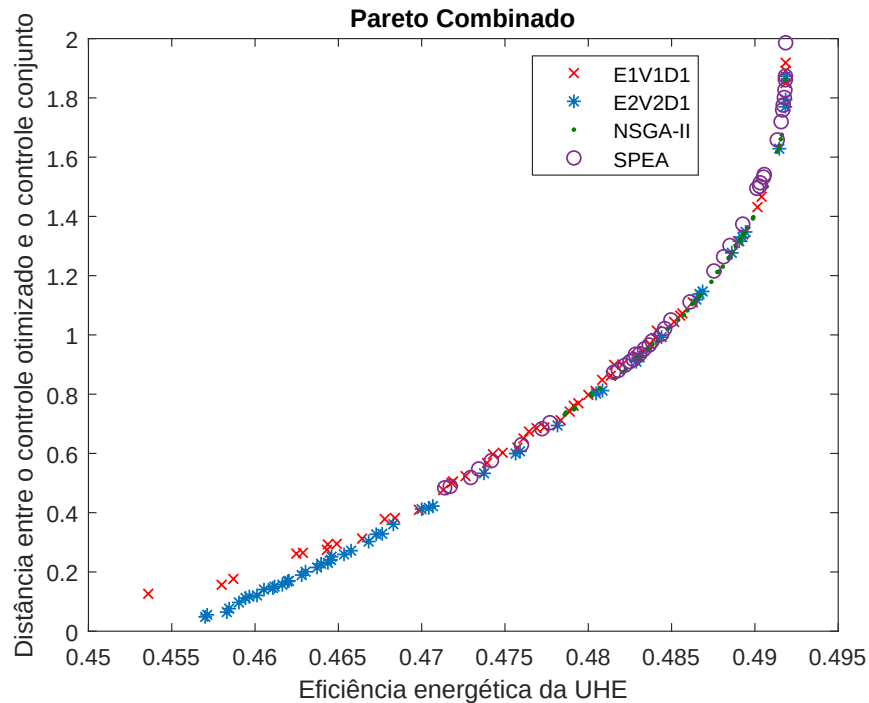
O primeiro problema real avaliado é o modelo de despacho elétrico de [Marcelino et al. \(2015\)](#), que maximiza a eficiência de uma UHE e minimiza a distancia entre os modos controle otimizado e conjunto. Os parâmetros utilizados no modelo deste trabalho são: demanda de $320MW$ e queda bruta de $54m$.

A [Figura 23](#) apresenta o pareto combinado contendo as 50 melhores soluções encontradas pelos algoritmos em suas 30 execuções. Nesta figura verifica-se que de maneira geral os três algoritmos obtiveram paretos semelhantes, as configurações do MESH se destacam com paretos mais longos e indicando uma maior diversidade. Próximo à origem, nota-se que algumas soluções da configuração E1V1D1 se encontram em uma posição em que são dominadas por outras soluções da configuração E2V2D1.

Como este problema é real e de pareto ótimo desconhecido, a análise estatística é feita apenas sobre a métrica de hipervolume dos paretos. A [Figura 24](#) apresenta o gráfico *boxplot* do hipervolume dos três algoritmos. O SPEA2 possui a maior caixa dentre os algoritmos com quatro *outliers* e possui os menores hipervolume dentre os algoritmos; as caixas do NSGA-II e da configuração E1V1D1 possuem tamanhos semelhantes e um *outlier*, sendo a caixa do E1V1D1 a com hipervolumes superiores; por fim, a configuração E2V2D1 apresenta a menor caixa com quatro *outliers* e os maiores hipervolume dentre os algoritmos.

Analisando as caixas do gráfico *boxplot*, não há interseções entre os três algorit-

Figura 23 – Pareto combinado dos algoritmos com o modelo de distância entre os controles otimizado e conjunto. Os números indicam o tamanho dos paretos dos algoritmos.



mos indicando que as configurações do MESH geram soluções com um maior hipervolume em relação ao NSGA-II e SPEA2. Entretanto, o tamanho do pareto combinado da configuração E1V1D1 sugere que este não gera paretos tão completos quanto os dos demais algoritmos. Para validar a análise anterior, a ANOVA com o p-valor de $1,87038e^{-19}$ confirma a diferença estatística significativa entre os algoritmos. O teste de Tukey da [Figura 25](#) detalha a diferença entre os algoritmos, demonstrando que as configurações do MESH geram soluções com hipervolumes maiores mas sem uma distinção entre elas.

Figura 24 – *Boxplot* dos hipervolumes dos três algoritmos no modelo de distância entre os controles otimizado e conjunto.

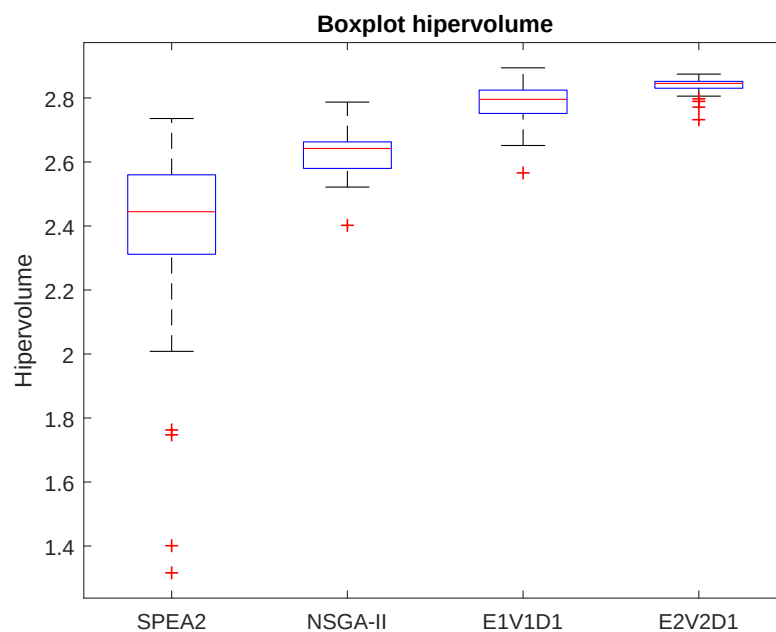
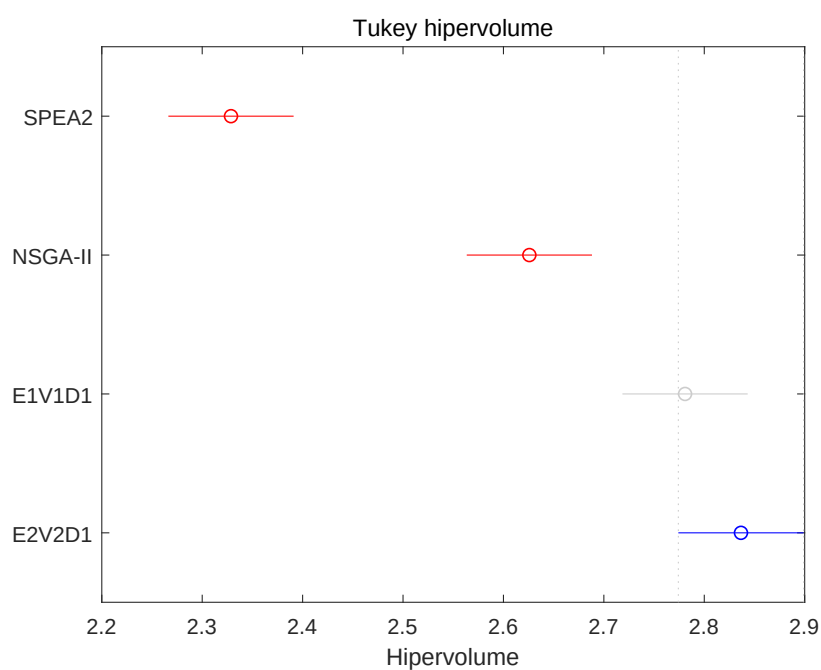


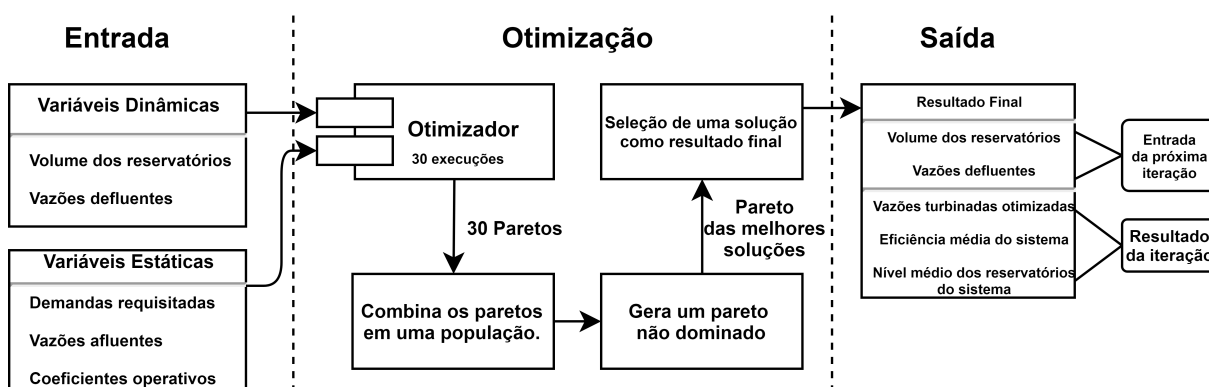
Figura 25 – Teste de Tukey dos hipervolumes dos três algoritmos no modelo de distância entre os controles otimizado e conjunto.



6.4.2 Estrutura para simulação do modelo de cascata proposto

A estratégia de reinicialização foi aplicada para tratar a otimização dinâmica do modelo proposto pelos três algoritmos avaliados, logo que sempre que o modelo é alterado pelo tempo, uma nova otimização será realizada. As simulações do modelo de UHE em cascata neste experimento têm o intervalo de tempo de uma hora, no total 24 horas serão simuladas. Cada iteração da simulação corresponde a uma das 24 horas e segue a sequência de passos explicitada no diagrama da [Figura 26](#). A escolha do intervalo de tempo de uma hora, é motivada pelos dados reais utilizados na simulação, que também adotam o mesmo intervalo.

Figura 26 – Diagrama para a simulação de uma iteração do modelo.



Cada iteração recebe dois tipos de variáveis de entrada. O conjunto das variáveis estáticas é definido antes do início da simulação e seus valores são independentes entre as iterações. Já as variáveis dinâmicas são transmitidas de uma iteração para a seguinte, exceto para a primeira iteração que também tem suas variáveis dinâmicas definidas antes do início da simulação, simbolizando o estado inicial do sistema. As variáveis de entrada são utilizadas em 30 execuções independentes de um algoritmo otimizador. Ao término de todas as execuções, é gerado um pareto combinado contendo as melhores soluções das 30 execuções.

A partir do pareto combinado, uma única solução é selecionada como resultado final da iteração. Em problemas MO, a escolha de uma solução é denominada de tomada de decisão e incorpora diferentes modelos, técnicas e muitas vezes um especialista no tema do problema. Neste trabalho, a solução selecionada é aquela que se encontra mais próxima do centro da curva do pareto. Com a solução final escolhida, os estados dos reservatórios e das vazões defluentes são transmitidos para a próxima iteração como variáveis de entrada dinâmicas.

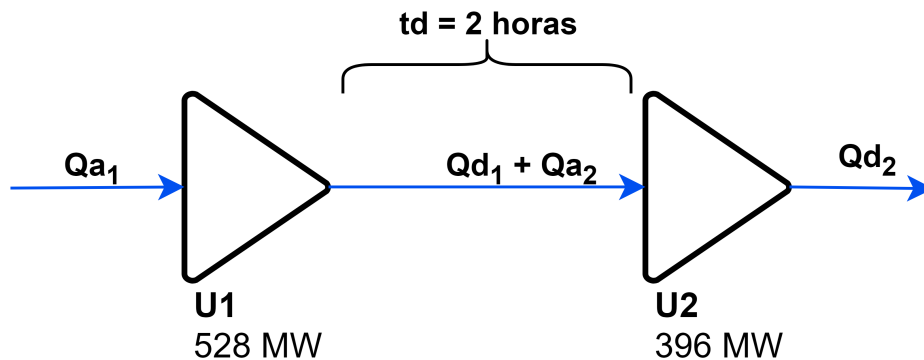
6.4.3 Descrição do sistema de usinas em cascata e suas variáveis para simulação

Por falta de acesso aos dados de um sistema real de usinas em cascata, foi criado um cenário genérico de duas UHE em cascata para a simulação. Ambas UHE deste cenário são fundamentadas na usina modelada no trabalho de [Marcelino \(2012\)](#) e em dados reais fornecidos pela empresa concessionária da usina. A usina abordada por [Marcelino \(2012\)](#) possui 6 unidades geradoras instaladas, porém sua estrutura hidráulica permite a operação de 8 unidades geradoras. O sistema de usinas usado na simulação é composto de uma UHE $U1$ de capacidade máxima de 528 MW com 8 unidades geradoras e outra UHE $U2$ que está jusante da $U1$ com capacidade máxima de 396 MW e 6 unidades geradoras.

Os reservatórios das duas UHE são idênticos e possuem o volume máximo de 19528 hm^3 e o volume mínimo de 4250 hm^3 . O volume inicial para ambos reservatórios é de 80%, que representa um cenário robusto em que há uma boa disponibilidade de água no reservatório e a altura da queda bruta garante um bom rendimento para as unidades geradoras.

Como a UHE que está sendo usada como referência nesta simulação não possui instrumentos para medição da evaporação, os termos relativos à evaporação são desconsiderados no balanço hídrico dos reservatórios. Por não ter uma outra UHE a jusante de $U1$, os termos de vazão defluente ($Q_{t_w,td}$ e $Q_{v_w,td}$) são nulos para o balanço hídrico de $U1$. Já no balanço hídrico de $U2$ considera-se que o tempo td de deslocamento da água entre $U1$ e $U2$ é de 2 horas. A [Figura 27](#) apresenta o diagrama do sistema com as UHE e as principais vazões.

Figura 27 – Diagrama do sistema de UHE em cascata usado na simulação.



Os coeficientes para os polinômios da cota montante e jusante são apresentados na [Tabela 9](#), como os reservatórios das duas UHEs são iguais, ambas usam dos mesmos coeficientes. Todas unidades geradoras são do mesmo tipo e suas restrições operativas são as mesmas apresentadas por [Marcelino \(2012\)](#). Os limites operativos das unidades geradoras são: $[35 - 65] MW$ para a potência, $[70 - 140] m^3 \cdot s^{-1}$ para a vazão. Os coeficientes

operativos ($\rho_{0j} \dots \rho_{5j}$) das unidades geradoras são os mesmos definidos na [Tabela 1](#) da [Subseção 2.3.2](#).

Tabela 9 – Coeficientes dos polinômios das cotas montante e jusante.

Cota montante				
a_0	a_1	a_2	a_3	a_4
$5,30e^{+02}$	$6,08e^{-03}$	$-4,84e^{-07}$	$2,20e^{-11}$	$-3,85e^{-16}$
Cota jusante				
b_0	b_1	b_2	b_3	b_4
$5,15e^{+02}$	$1,61e^{-03}$	$-2,55e^{-07}$	$2,89e^{-11}$	$-1,18e^{-15}$

Fonte: ([BRANDÃO, 2004](#))

A partir dos dados de um documento de restrições operativas fornecido pela empresa concessionária, os limites das vazões defluentes são definidos no intervalo de $[400 - 2500]m^3s^{-1}$. Apesar de existir um limite máximo para a vazão vertida, optou-se por usar o mesmo limite máximo da vazão defluente, pois o limite operativo para a vazão vertida é superior ao limite da vazão defluente.

Foi fornecida por uma empresa concessionária uma base de dados contendo as medições de vazões afluentes e a potência gerada por uma UHE, para o período de 12/07/2011 até 11/08/2011. A partir dessa base de dados, foram selecionados os dias 26/07/2011 e 27/07/2011 para serem respectivamente as vazões afluentes para as usinas $U1$ e $U2$. Quanto às demandas de energia, considerou-se os valores referentes ao dia 26/07/2011 para a usina $U2$ e para a usina $U1$ foi adicionado 25% sobre os valores de referência, uma vez que $U1$ possui 8 unidades geradoras.

Os valores das vazões afluentes são apresentados pela [Tabela 10](#). Como a soma das vazões afluentes é inferior a soma das vazões defluentes das UHE, a tendência do nível médio do sistema é de diminuir ao longo do dia. Porém é importante ressaltar que usina $U1$ eleva o nível do reservatório da usina $U2$, pois a usina $U1$ possui um número maior de unidades geradoras e sua defluência afeta o reservatório de $U2$. As demandas de energia são apresentadas pela [Tabela 11](#).

Tabela 10 – Vazões afluentes para a simulação.

Vazões afluentes em m^3s^{-1}					
Hora	U1	U2	Hora	U1	U2
0	102,83	208,30	12	341,35	228,87
1	102,65	289,21	13	341,00	114,85
2	103,22	297,96	14	340,62	114,86
3	105,00	192,64	15	340,30	114,87
4	221,29	201,45	16	339,41	114,81
5	223,18	204,59	17	227,74	114,80
6	110,80	218,98	18	236,17	230,89
7	227,00	227,18	19	245,00	229,30
8	114,06	343,05	20	253,80	115,34
9	228,11	343,05	21	376,80	115,33
10	227,75	228,87	22	376,81	109,53
11	227,45	235,15	23	385,52	223,95

Tabela 11 – Demandas de energia para a simulação.

Demandas de energia em MW					
Hora	U1	U2	Hora	U1	U2
0	330	264	12	292	234
1	330	264	13	300	240
2	330	264	14	292	234
3	322	258	15	300	240
4	322	258	16	292	234
5	330	264	17	292	234
6	330	264	18	420	336
7	330	264	19	437	343
8	300	240	20	437	343
9	292	234	21	437	343
10	292	234	22	437	343
11	292	234	23	445	349

6.4.4 Avaliação das soluções para o modelo de cascata

Para o modelo de UHE em cascata, primeiramente são avaliadas as soluções que os algoritmos fornecem ao longo da simulação da operação das UHE em 24 horas, conforme elucidado na [Subseção 6.4.2](#). Os resultados apresentados são as soluções que foram escolhidas ao fim das iterações da simulação. Portanto, a qualidade dos resultados não necessariamente reflete a capacidade dos algoritmos, já que as comparações são feitas com apenas uma solução encontrada por cada algoritmo e não com o conjunto de soluções do pareto.

A [Tabela 12](#) e a [Tabela 13](#) apresentam os valores das duas funções objetivos das soluções escolhidas em cada algoritmo e também os valores relativos ao controle conjunto. Apesar da diferença percentual do volume dos reservatórios no estado inicial de 80% e os demais valores encontrados na simulação estar na ordem de 10^{-2} , esta pequena diferença numérica representa milhões de litros de água. Ambas tabelas destacam as soluções que dominam o controle conjunto ou são dominadas por ele. Ao longo da simulação, a configuração E1V1D1 tem 12 iterações em que a sua solução escolhida é inferior ao controle conjunto, sendo assim o algoritmo que é dominado mais vezes. Já a configuração E2V2D1 é o algoritmo que domina mais vezes o controle conjunto em um total de 17 iterações, além disso nenhuma de suas soluções é dominada pelo controle conjunto.

Dentre os algoritmos clássicos, o SPEA2 tem a segunda maior quantidade de soluções dominadas pelo controle conjunto com 6 soluções dominadas. O SPEA2 é o algoritmo com a menor quantidade de soluções que dominam o controle conjunto, com um total de 4 soluções. Assim como a configuração E2V2D1, o NSGA-II não tem nenhuma solução dominada pelo controle conjunto e também mantém a segunda maior quantidade de soluções que dominam o controle conjunto, com um total de 12 iterações com soluções superiores ao controle conjunto. A partir da hora 18, há um aumento na demanda de energia para ambas UHE, nestas horas simuladas temos todos os algoritmos sem soluções e a maioria deles dominam o controle conjunto. Com esta avaliação, verificou-se que é vantajoso utilizar o modelo de geração de energia em UHEs em cascata proposto deste trabalho, sendo possível tornar o sistema de usinas mais eficiente que o controle conjunto.

Tabela 12 – Valores das funções objetivo das soluções escolhidas, no qual:

 F_1 - Eficiência energética média do sistema em MW por m^3s^{-1} F_2 - Nível médio dos reservatórios do sistema em %.Soluções com o símbolo $\uparrow$ dominam o controle conjunto.Soluções com o símbolo $\downarrow$ são dominadas pelo controle conjunto.

Valores das funções objetivo das soluções escolhidas.						
Hora	Função	E1V1D1	E2V2D1	SPEA2	NSGA-II	Conjunto
0	F_1	0,468713	$\uparrow$ 0,468734	$\downarrow$ 0,468706	0,468720	0,468729
	F_2	79,994256	$\uparrow$ 79,994256	$\downarrow$ 79,994251	79,994260	79,994255
1	F_1	0,468679	0,468701	$\downarrow$ 0,468681	0,468704	0,468711
	F_2	79,986189	79,986204	$\downarrow$ 79,986168	79,986201	79,986181
2	F_1	$\downarrow$ 0,468691	0,468734	0,468695	$\uparrow$ 0,468710	0,468708
	F_2	$\downarrow$ 79,984660	79,984871	79,984709	$\uparrow$ 79,984715	79,984703
3	F_1	0,468274	0,468338	0,468291	0,468313	0,468320
	F_2	79,982646	79,982497	79,982562	79,982571	79,982536
4	F_1	$\downarrow$ 0,468299	$\uparrow$ 0,468328	0,468241	$\uparrow$ 0,468321	0,468317
	F_2	$\downarrow$ 79,981367	$\uparrow$ 79,981965	79,981572	$\uparrow$ 79,981552	79,981523
5	F_1	$\downarrow$ 0,468685	$\uparrow$ 0,468717	0,468660	0,468684	0,468697
	F_2	$\downarrow$ 79,979803	$\uparrow$ 79,980358	79,980175	79,980182	79,980139
6	F_1	$\downarrow$ 0,468656	$\uparrow$ 0,468716	0,468637	0,468688	0,468692
	F_2	$\downarrow$ 79,977677	$\uparrow$ 79,978194	79,977883	79,977894	79,977851
7	F_1	$\downarrow$ 0,468668	$\uparrow$ 0,468703	0,468638	0,468680	0,468690
	F_2	$\downarrow$ 79,976379	$\uparrow$ 79,977637	79,976897	79,976911	79,976862
8	F_1	$\downarrow$ 0,466848	$\uparrow$ 0,466889	0,466859	0,466879	0,466881
	F_2	$\downarrow$ 79,976598	$\uparrow$ 79,977602	79,976966	79,976971	79,976921
9	F_1	$\downarrow$ 0,466264	$\uparrow$ 0,466282	0,466226	$\uparrow$ 0,466279	0,466270
	F_2	$\downarrow$ 79,977970	$\uparrow$ 79,979023	79,978339	$\uparrow$ 79,978356	79,978293
10	F_1	$\downarrow$ 0,466248	$\uparrow$ 0,466298	0,466220	$\uparrow$ 0,466284	0,466269
	F_2	$\downarrow$ 79,977737	$\uparrow$ 79,978684	79,978072	$\uparrow$ 79,978109	79,978048
11	F_1	0,466259	0,466267	0,466239	$\uparrow$ 0,466276	0,466268
	F_2	79,977733	79,977988	79,977751	$\uparrow$ 79,977758	79,977709
12	F_1	$\downarrow$ 0,466265	$\uparrow$ 0,466294	$\downarrow$ 0,466230	$\uparrow$ 0,466270	0,466270
	F_2	$\downarrow$ 79,978304	$\uparrow$ 79,978823	$\downarrow$ 79,978353	$\uparrow$ 79,978390	79,978361
13	F_1	$\downarrow$ 0,466866	0,466867	0,466837	0,466867	0,466883
	F_2	$\downarrow$ 79,977588	79,977752	79,977703	79,977716	79,977697
14	F_1	$\downarrow$ 0,466246	0,466262	$\downarrow$ 0,466226	0,466283	0,466267
	F_2	$\downarrow$ 79,977169	79,977777	$\downarrow$ 79,977270	79,977274	79,977292
15	F_1	0,466864	$\uparrow$ 0,466895	$\downarrow$ 0,466840	0,466884	0,466881
	F_2	79,976830	$\uparrow$ 79,977503	$\downarrow$ 79,976719	79,976752	79,976772
16	F_1	$\downarrow$ 0,466250	$\uparrow$ 0,466270	0,466241	0,466286	0,466265
	F_2	$\downarrow$ 79,976304	$\uparrow$ 79,977762	79,976429	79,976308	79,976356

Tabela 13 – Valores das funções objetivo das soluções escolhidas, no qual:
 F_1 - Eficiência energética média do sistema em MW por m^3s^{-1}
 F_2 - Nível médio dos reservatórios do sistema em %.
 Soluções com o símbolo $\uparrow$ dominam o controle conjunto.
 Soluções com o símbolo $\downarrow$ são dominadas pelo controle conjunto.

Valores das funções objetivo das soluções escolhidas.						
Hora	Função	E1V1D1	E2V2D1	SPEA2	NSGA-II	Conjunto
17	F_1	0,466225	0,466259	$\downarrow$ 0,466233	0,466280	0,466262
	F_2	79,975091	79,976736	$\downarrow$ 79,974983	79,975011	79,975061
18	F_1	$\uparrow$ 0,468451	$\uparrow$ 0,468467	0,468467	$\uparrow$ 0,468461	0,468387
	F_2	$\uparrow$ 79,970347	$\uparrow$ 79,973476	79,970265	$\uparrow$ 79,970294	79,970287
19	F_1	$\uparrow$ 0,467703	$\uparrow$ 0,467775	$\uparrow$ 0,467678	$\uparrow$ 0,467734	0,467636
	F_2	$\uparrow$ 79,965208	$\uparrow$ 79,973048	$\uparrow$ 79,965132	$\uparrow$ 79,965150	79,965083
20	F_1	$\uparrow$ 0,467709	$\uparrow$ 0,467755	$\uparrow$ 0,467732	$\uparrow$ 0,467722	0,467627
	F_2	$\uparrow$ 79,961549	$\uparrow$ 79,972079	$\uparrow$ 79,961478	$\uparrow$ 79,961494	79,961386
21	F_1	$\uparrow$ 0,467705	$\uparrow$ 0,467786	$\uparrow$ 0,467712	$\uparrow$ 0,467746	0,467622
	F_2	$\uparrow$ 79,959370	$\uparrow$ 79,973214	$\uparrow$ 79,959305	$\uparrow$ 79,959322	79,959174
22	F_1	$\uparrow$ 0,467679	$\uparrow$ 0,467745	$\uparrow$ 0,467700	$\uparrow$ 0,467693	0,467616
	F_2	$\uparrow$ 79,957391	$\uparrow$ 79,973156	$\uparrow$ 79,957079	$\uparrow$ 79,957096	79,956910
23	F_1	$\uparrow$ 0,467173	$\uparrow$ 0,467176	0,467009	$\uparrow$ 0,467124	0,467021
	F_2	$\uparrow$ 79,955743	$\uparrow$ 79,974000	79,955692	$\uparrow$ 79,955711	79,955484

6.4.5 Avaliação do desempenho dos algoritmos no modelo de cascata

A segunda avaliação para o modelo de cascata compara o desempenho e o comportamento dos algoritmos durante a simulação do modelo. A [Figura 28](#), [Figura 29](#), [Figura 30](#) e [Figura 31](#) apresentam os paretos combinados de cada hora da simulação, incluindo o ponto de referência do controle conjunto e a área que delimita a região de soluções dominadas pelo controle conjunto.

Na primeira hora da simulação, o pareto combinado da configuração E2V2D1 é o mais distante da origem sugerindo que esta configuração gera melhores soluções e é o único que não possui nenhum ponto dominado pelo controle conjunto. Em seguida o NSGA-II é o segundo pareto mais distante, com um único ponto dominado pelo controle conjunto. Os paretos do SPEA2 e do E1V1D1 possuem mais pontos dominados pelo controle conjunto, sendo o SPEA2 o com a maior quantidade. Dentre estes dois algoritmos, a posição do pareto do E1V1D1 sugere um melhor desempenho dele em relação ao SPEA2.

Desde a hora 1 até a hora 17, os paretos dos algoritmos seguem um padrão: o pareto da configuração E2V2D1 é o mais distante, seguido pelo NSGA-II, depois o SPEA2 e por último a configuração E1V1D1. O E1V1D1 é o único que permanece com uma parte de seu pareto dominado durante todas as horas deste intervalo. Já o E2V2D1 não é dominado pelo controle conjunto em nenhuma hora de toda a simulação. O SPEA2 e o NSGA-II de forma intermitente são dominados pelo controle conjunto, sendo o SPEA2 dominado mais vezes do que o NSGA-II.

Após a hora 18, devido ao aumento da demanda de energia nas UHE o espaço de busca que possui soluções factíveis é reduzido, com isto os algoritmos tem uma maior dificuldade em gerar paretos completos. Como consequência o tamanho dos paretos combinados é reduzido drasticamente havendo menos de 6 pontos em todos algoritmos em todas as horas deste intervalo. Nenhum dos algoritmos gera soluções dominadas pelo controle conjunto. Os pontos encontrados pelo E2V2D1 se mostram mais vantajosos em relação aos pontos dos outros algoritmos por manter o nível dos reservatórios mais alto. Os outros algoritmos têm níveis de reservatórios semelhantes, havendo uma maior variação entre suas eficiências energéticas.

Pelo o exposto nos gráficos de paretos combinados, a configuração E2V2D1 aparenta ter o melhor dentre os algoritmos comparados e a configuração E1V1D1 o pior desempenho. Dentre os dois algoritmos clássicos, o NSGA-II aparenta ter um melhor desempenho em relação ao SPEA2. Ao se comparar estes gráficos com a [Tabela 12](#) e a [Tabela 13](#) das soluções escolhidas, fica evidente que todos os algoritmos são capazes de gerar soluções que não são dominadas pelo controle conjunto e a importância da tomada de decisão no pareto combinado.

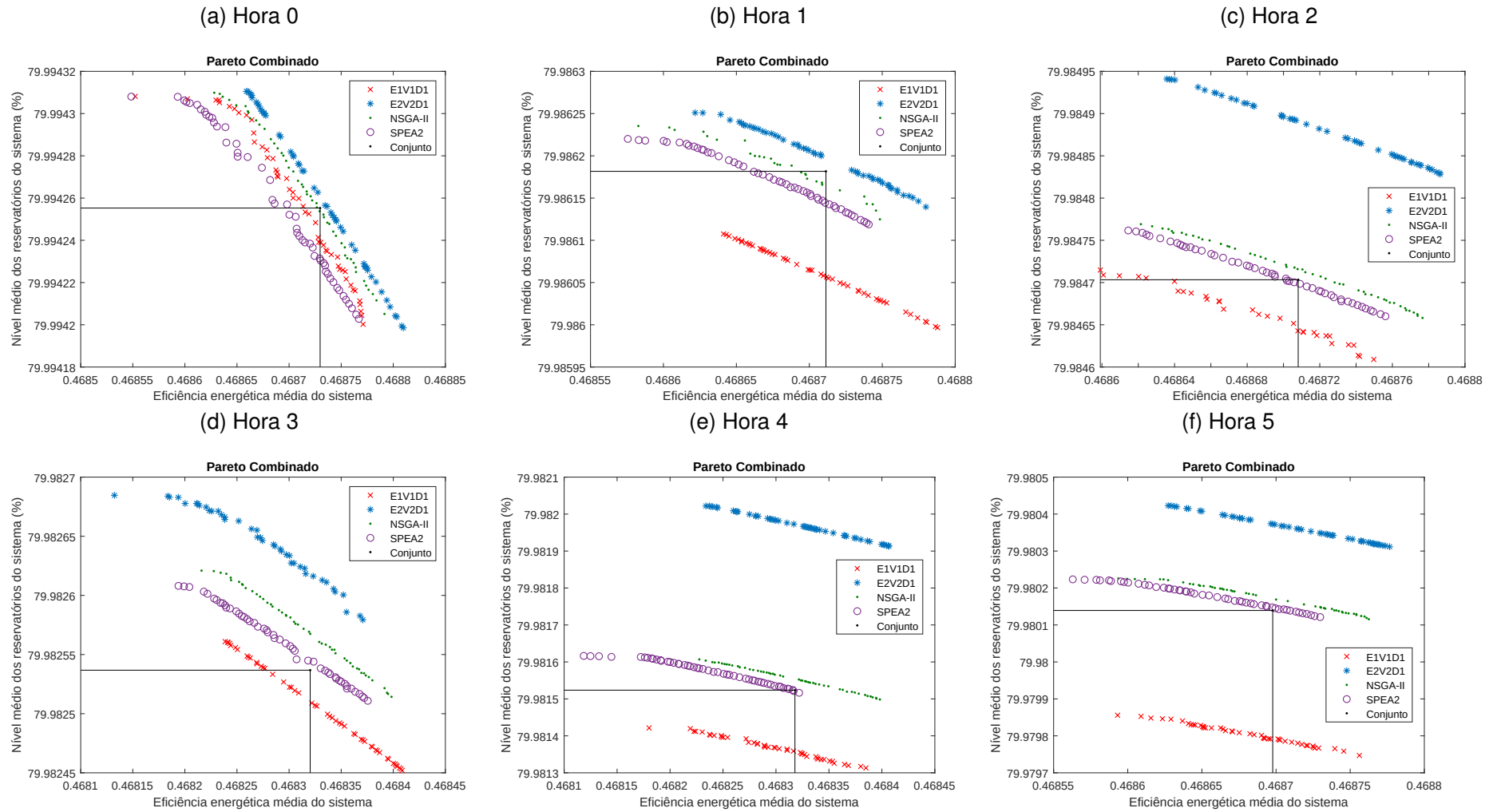


Figura 28 – Pareto combinado dos algoritmos no modelo de cascata nas horas 0-5.

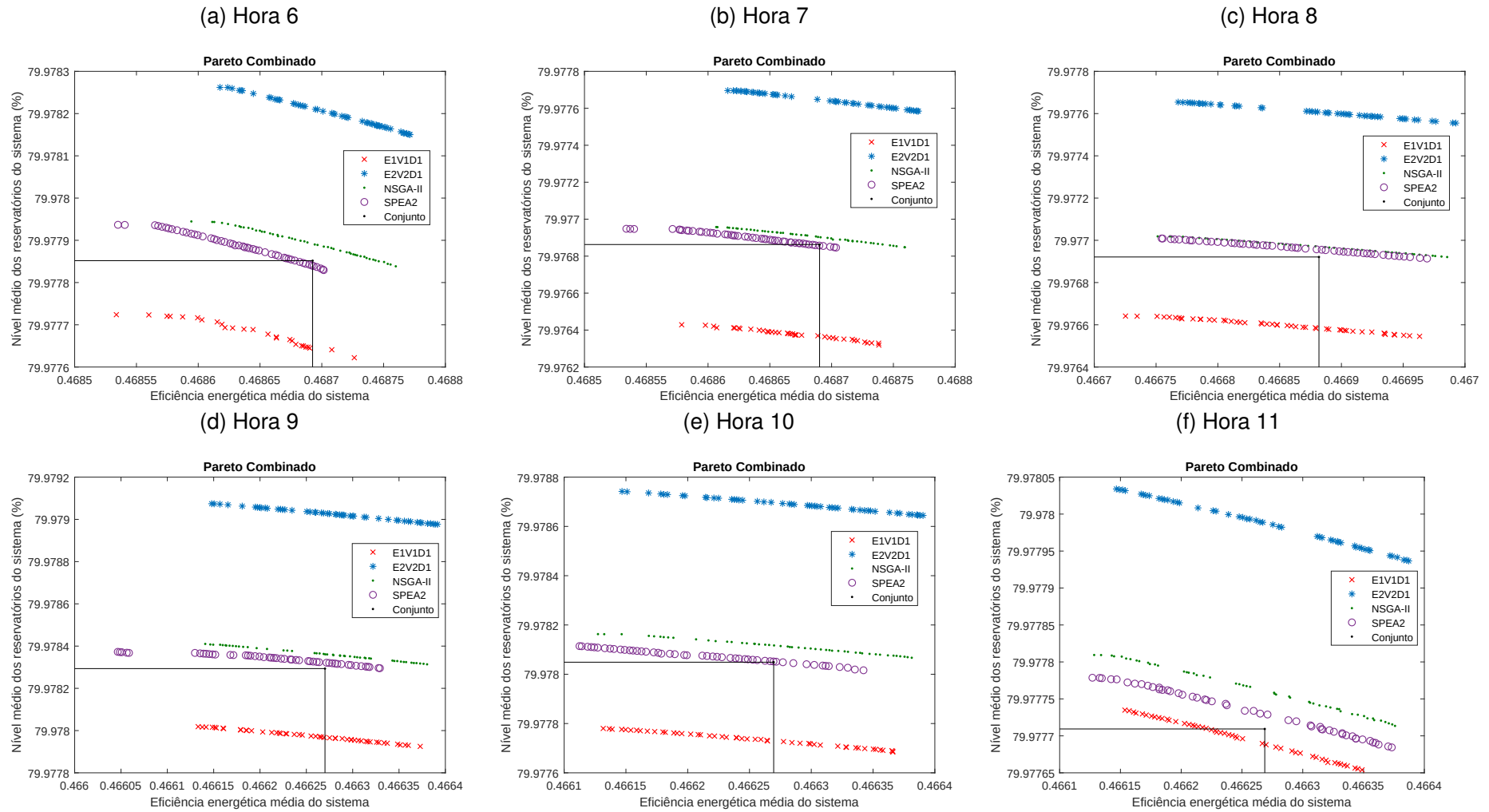


Figura 29 – Pareto combinado dos algoritmos no modelo de cascata nas horas 6-11.

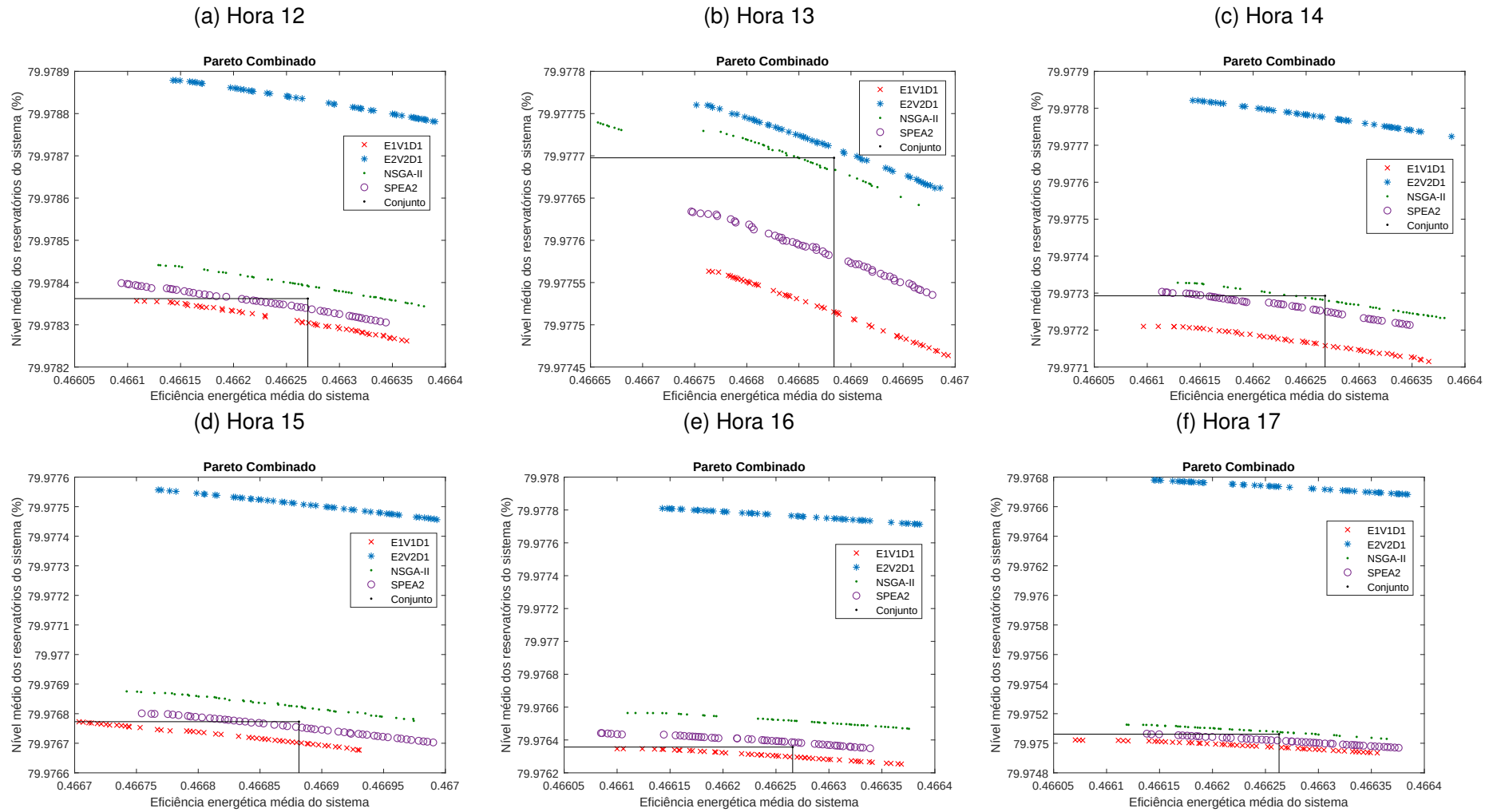


Figura 30 – Pareto combinado dos algoritmos no modelo de cascata nas horas 12-17.

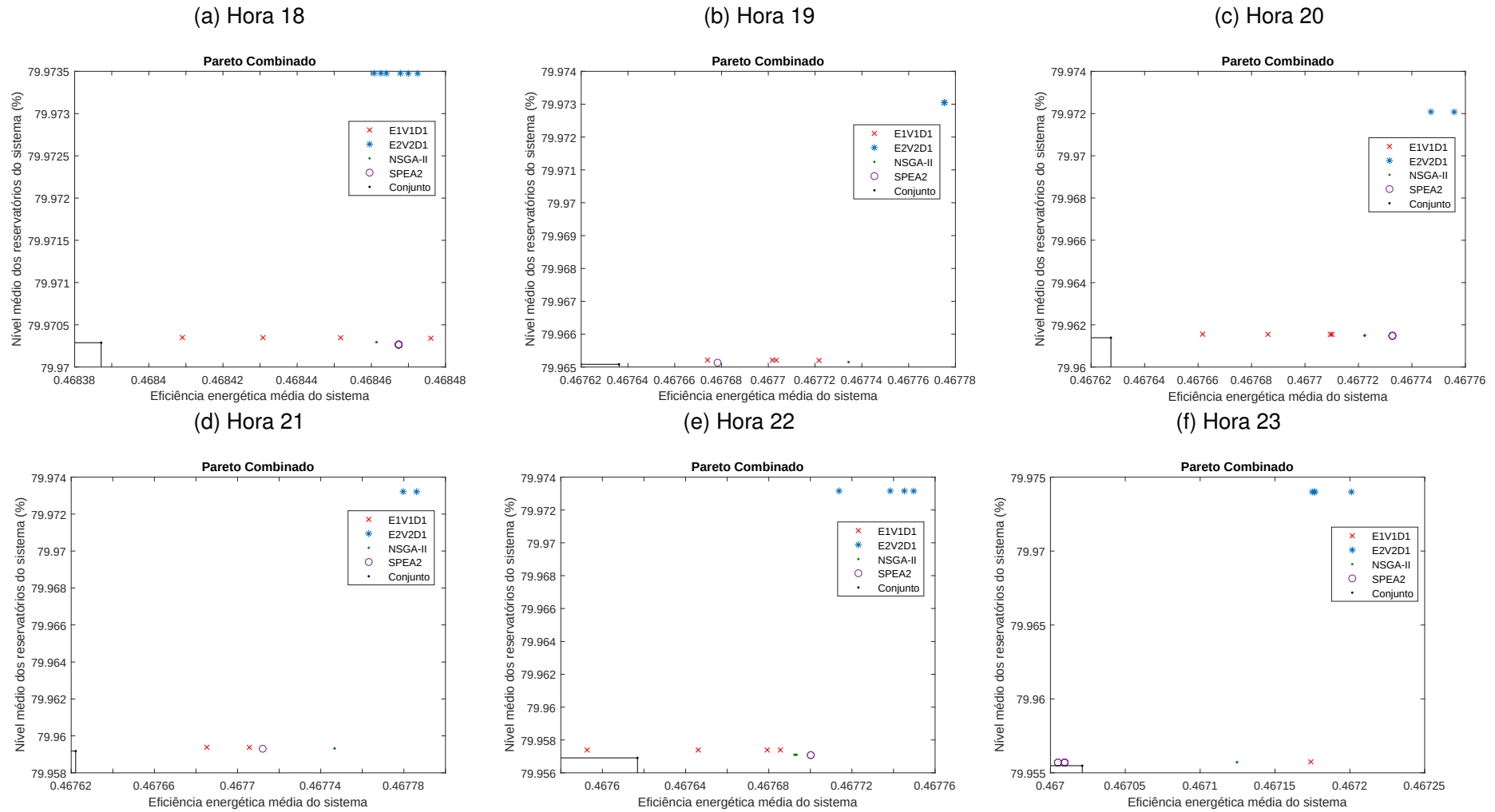


Figura 31 – Pareto combinado dos algoritmos no modelo de cascata nas horas 18-23.

A primeira hora da simulação é a única iteração em que todos os algoritmos possuem os mesmos estados iniciais e que, portanto, estão otimizando o modelo em condições idênticas. O gráfico *boxplot* dos hipervolumes dos algoritmos é apresentado pela [Figura 32](#). A caixa da configuração E2V2D1 é a única que não possui interseções com as demais e é também a de menor tamanho e maior hipervolume, logo pode-se afirmar que esta configuração gera as melhores soluções. As caixas dos outros algoritmos possuem inserções entre si, logo não possível diferenciá-los por este gráfico.

A ANOVA com o p-valor de $1,7829e^{-15}$ confirma que há diferenças significativas entre os algoritmos. O teste de Tukey da [Figura 33](#), classifica os algoritmos quanto aos seus hipervolumes, dando a liderança para a configuração E2V2D1. O NSGA-II tem o segundo melhor desempenho, enquanto o SPEA2 tem o terceiro e a configuração E1V1D1 tem o pior desempenho.

As análises dos hipervolumes das demais horas da simulação, são sintetizadas na [Tabela 14](#). Os resultados dos hipervolumes, confirmam a análise feita com os paretos combinados. A configuração E2V2D1 tem os maiores hipervolumes em todas as horas simuladas. O NSGA-II tem a segunda melhor performance, estando empatado com o SPEA2 nas horas 7 e 8. A configuração E1V1D1 tem os piores hipervolumes durante toda a simulação, exceto para a primeira hora.

Figura 32 – *Boxplot* dos hipervolumes dos três algoritmos no modelo de cascata para a primeira hora de simulação.

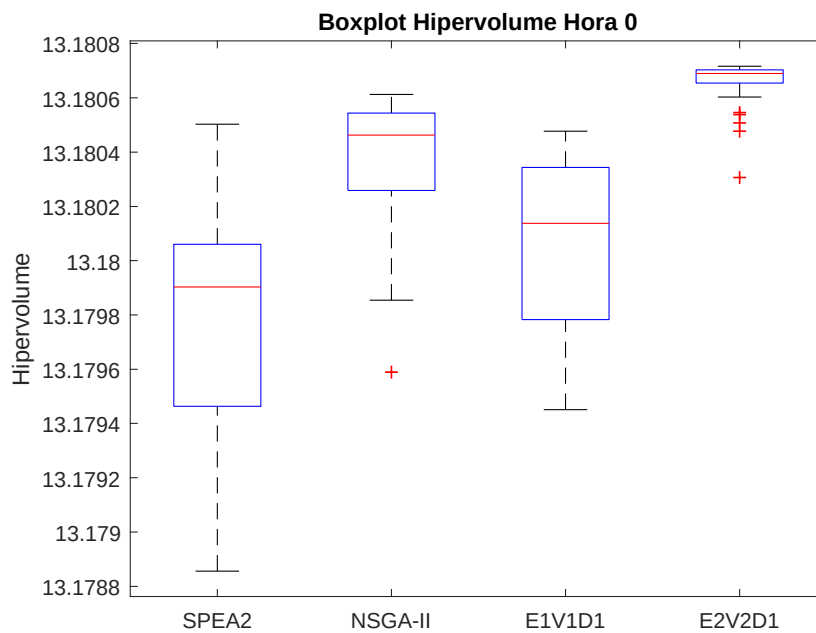


Figura 33 – Teste de Tukey dos hipervolumes dos três algoritmos no modelo de cascata para a primeira hora de simulação.

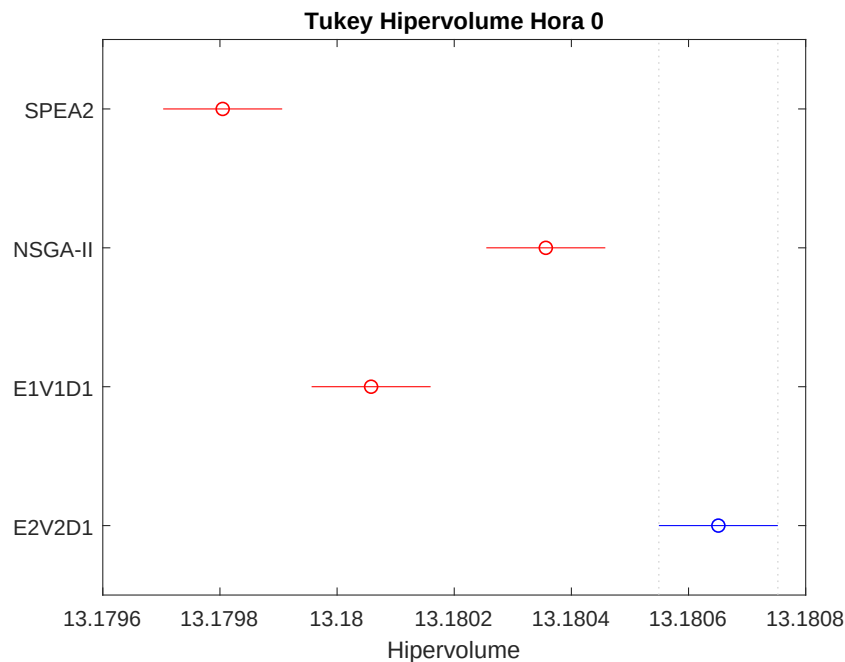


Tabela 14 – Análise dos hipervolumes dos algoritmos no modelo de cascata.

Hora	P-valor ANOVA	Classificação Tukey			
		1º	2º	3º	4º
0	$1,7829e^{-15}$	E2V2D1	NSGA-II	E1V1D1	SPEA2
1	$1,5665e^{-17}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
2	$1,8762e^{-16}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
3	$2,1249e^{-18}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
4	$3,8165e^{-15}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
5	$4,8913e^{-11}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
6	$1,4564e^{-13}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
7	$2,1125e^{-8}$	E2V2D1	NSGA-II SPEA2	E1V1D1	—
8	$2,3365e^{-7}$	E2V2D1	NSGA-II SPEA2	E1V1D1	—
9	$1,6233e^{-10}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
10	$3,2343e^{-12}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
11	$2,7785e^{-13}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
12	$2,6663e^{-14}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
13	$4,2678e^{-14}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
14	$4,3698e^{-13}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
15	$3,2253e^{-8}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
16	$1,7441e^{-9}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
17	$3,1114e^{-9}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
18	$1,7866e^{-19}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
19	$1,1122e^{-19}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
20	$1,3356e^{-18}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
21	$1,8897e^{-19}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
22	$1,2251e^{-18}$	E2V2D1	NSGA-II	SPEA2	E1V1D1
23	$1,2512e^{-19}$	E2V2D1	NSGA-II	SPEA2	E1V1D1

6.4.6 Considerações finais

Os objetivos do experimento 3 compreendiam avaliar o desempenho do MESH em problemas reais e validar se o uso do modelo proposto de geração de energia elétrica com UHEs em cascata é vantajoso. Para o modelo proposto foi elaborado uma estrutura para simular a operação de um sistema de UHEs em cascata por 24 horas e baseado em dados reais de uma UHEs criou-se um cenário genérico com duas usinas. Além do modelo proposto neste trabalho, o modelo de [Marcelino et al. \(2015\)](#) foi utilizado para avaliar os algoritmos. No primeiro modelo avaliado, os paretos combinado dos algoritmos quase se coincidem, sendo a principal diferença na extensão deles, em que as configurações do MESH possuem paretos mais longos, sendo o da configuração E1V1D1 mais espargo. A análise estatística dos hipervolumes mostra que ambas configurações do MESH geram soluções melhores do que as do NSGA-II e SPEA2.

Ao comparar os paretos combinados das horas simuladas do modelo proposto neste trabalho com a tabela de soluções escolhidas, fica evidente que é necessário uma nova estratégia para a tomada de decisão, com o objetivo de se evitar que soluções dominadas pelo controle conjunto sejam escolhidas. Os gráficos dos paretos combinados demonstram que o modelo proposto se torna de otimização mais difícil, quando a demanda de energia sobre as UHE é incrementada.

Avaliando as duas funções objetivo das soluções dos algoritmos, pode ser afirmado que o uso do modelo é capaz de tornar o sistema das usinas em cascata mais eficiente tanto na geração de energia, quanto no nível médio dos reservatórios. Este experimento mostrou que as configurações do MESH são sensíveis ao problema a ser otimizado, pois diferente do que foi visto nas funções ZDT e no primeiro modelo avaliado, apenas a configuração E2V2D1 teve um desempenho superior aos algoritmos SPEA2 e NSGA-II no modelo com UHE em cascata.

7 Conclusão e trabalhos futuros

7.1 Conclusão

Neste trabalho foi desenvolvido um novo algoritmo MO híbrido denominado MESH, que se fundamenta no funcionamento do algoritmo mono-objetivo híbrido C-DEEPSO. Uma revisão bibliográfica foi realizada para identificar diferentes estratégias de algoritmos MO híbridos similares ao C-DEEPSO e de abordagens MO para os algoritmos DE e PSO. A partir desta revisão incorporou-se diferentes operadores e técnicas ao funcionamento do C-DEEPSO para a criação do algoritmo MESH.

O usuário do MESH necessita definir parâmetros para dois operadores. O operador de escolha do guia de enxame do MESH permite duas opções: uma no qual o guia é uma partícula pertencente à memória e outra é se baseia na fronteira seguinte à fronteira da partícula. No operador de mutação diferencial é necessário definir dois parâmetros: um para definir a origem do grupo para a amostragem dos vetores e outro para qual estratégia evolutiva do DE será aplicada. Na amostragem dos vetores, as possíveis opções são: população, memória ou combinação da população e memória. E cinco estratégias evolutivas do DE foram implementadas neste trabalho. No total há 30 possíveis configurações para o uso do MESH.

As 30 configurações do MESH foram avaliadas nas funções *benchmark* ZDT3, ZDT4 e ZDT6 no experimento 1 deste trabalho, com o objetivo de se identificar duas melhores configurações para serem utilizadas nos experimentos seguintes. Na função ZDT6 cinco configurações se mostraram equivalentes, as funções ZDT4 e ZDT3 tiveram duas configurações com desempenhos equivalentes em suas respectivas funções. Por estar presente dentre as melhores configurações na função ZDT6 e ZDT4 ao mesmo tempo, a configuração E1V1D1 foi a primeira escolha. Já a segunda escolha foi a E2V2D1 por ser aquela com maior diferença em relação as outras opções.

No segundo experimento, as duas configurações escolhidas do MESH foram avaliadas e comparadas com os dois algoritmos MO clássicos NSGA-II e SPEA2, nas funções *benchmark* ZDT1, ZDT2, ZDT3, ZDT4 e ZDT6. O MESH gerou melhores soluções que o NSGA-II e o SPEA2 em três das cinco funções ZDT, contudo deve-se ressaltar que na ZDT2 na maioria dos casos, o MESH não foi capaz de gerar um pareto completo, estando preso a um ponto do pareto ótimo. Nas outras duas funções ZDT, o MESH e o NSGA-II geram paretos sobre o pareto ótimo, contudo o MESH é rápido computacionalmente. Com isto considera-se que o MESH tem um desempenho superior aos dois algoritmos nas funções ZDT.

Neste trabalho também foi proposto um novo modelo matemático para a geração de energia com UHE em cascata. O modelo proposto aprimora o modelo mono-objetivo de despacho elétrico em uma UHE de [Marcelino \(2012\)](#) e [Marcelino \(2017\)](#), adicionando o balanço hídrico às usinas e tornando a altura da queda bruta uma variável dependente do sistema em vez de um parâmetro de entrada do modelo. O balanço hídrico permite acoplar o funcionamento das usinas em cascata e torna o modelo um sistema dinâmico, no qual é necessário incluir um estado inicial do sistema. Para utilizar o modelo proposto como estudo de caso na avaliação do desempenho do MESH em problemas reais, o modelo possui uma segunda função objetiva que visa maximizar o nível médio dos reservatórios do sistema de UHEs.

Para avaliar o modelo proposto, elaborou-se uma estrutura para simulação de 24 horas de operação de duas UHEs em cascata. Em cada iteração da simulação, 30 execuções de um algoritmo otimizam a operação de uma hora da simulação, os paretos das execuções são combinados e gera-se um pareto combinado das 30 execuções. Uma solução do pareto combinado é escolhida para que seu estado final seja o estado inicial da iteração seguinte da simulação. Os dados de entrada para simulação foram baseados em dados reais de uma UHE brasileira.

O terceiro experimento deste trabalho avaliou o desempenho do MESH em dois modelos de geração de energia de UHEs comparando-o com os algoritmos SPEA2 e NSGA-II. Em ambos modelos a configuração E2V2D1 do MESH demonstrou ser capaz de gerar melhores soluções que os algoritmos NSGA-II, SPEA2. A configuração E1V1D1, esta configuração por sua vez teve o pior desempenho com o modelo proposto. Nas avaliações com problemas reais, ficou evidente que o desempenho do MESH é sensível à combinação de sua configuração e do tipo de problema a ser otimizado.

O terceiro experimento também evidenciou que assim como no modelo de despacho elétrico de [Marcelino \(2012\)](#) e [Marcelino \(2017\)](#), o uso modelo de UHEs em cascata deste trabalho é vantajoso em relação ao controle conjunto, tanto para o objetivo de eficiência energética média, quanto para o objetivo do nível médio dos reservatórios. O objetivo geral deste trabalho é contribuir para a área de computação evolutiva, com um novo algoritmo MO híbrido e para o setor de geração de energia elétrica com um modelo de geração de energia de UHEs em cascata. Conforme exposto neste trabalho, considera-se que objetivo do mesmo foi cumprido, pelo desenvolvimento do algoritmo MESH e o modelo de geração energia com UHEs de cascata apresentado.

7.2 Trabalhos futuros

Como possíveis trabalhos futuros, sugere-se as seguintes direções:

- A avaliação completa das configurações do algoritmo MESH em outros problemas

reais e de *benchmark*. Apenas duas configurações foram avaliadas em problemas reais e ficou evidente que o desempenho do algoritmo é sensível à combinação de configuração e problema a ser otimizado, uma avaliação completa poderia esclarecer as vantagens de cada configuração;

- Aprimoramento do algoritmo MESH pela incorporação de operadores e técnicas de outros algoritmos;
- Comparar o desempenho do MESH com outros algoritmos MO híbridos;
- Propor novas estratégias para a tomada de decisão de forma a evitar que soluções inferiores ao controle conjunto sejam selecionadas no modelo proposto;
- Reavaliar a otimização do modelo de cascata proposto com demandas altas sobre as UHE e com maior tempo computacional para os algoritmos;
- Otimizar o modelo de cascata proposto com algoritmos evolutivos adaptados para a otimização dinâmica;
- Incorporar perdas mecânicas e elétricas ao modelo proposto;
- Aplicar o modelo proposto em um cenário real;
- Simular o modelo proposto com as vazões vertidas dos reservatórios como variáveis de decisão. As vazões vertidas têm efeitos principalmente sobre o segundo objetivo e em um cenário em que há UHEs do tipo fio d'água;
- Otimizar as usinas do modelo proposto de forma independente. Neste trabalho, as variáveis de decisão incorporam as vazões turbinadas de todas as usinas em um mesmo vetor a ser otimizado. Ao separar as vazões de cada usina em otimizações independentes, é possível que cada otimização seja de resolução mais fácil, havendo assim uma troca no custo computacional para resolver o modelo por soluções de maior qualidade;
- Avaliar se é vantajoso aplicar a demanda de energia sobre o sistema de UHE como um todo em substituição de se aplicar as demandas de forma individual para cada UHE do sistema. Aplicar a demanda de energia sobre o sistema, permite uma maior flexibilidade na escolha dos despachos de energia nas UHE.

Referências

- ARCE, A. **Despacho ótimo de unidades geradoras em sistemas hidrelétricos via heurística baseada em relaxação lagrangeana e programação dinâmica**. Tese (Doutorado) — UNICAMP, 2006. Citado na página 29.
- BERTSEKAS, D. P. **Dynamic Programming and Optimal Control**. 3. ed. [S.l.]: Athena Scientific, 2005. v. 1. Citado na página 8.
- BRANDÃO, J. L. B. **Modelo para operação de sistemas de reservatórios com usos múltiplos**. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, 2004. Citado na página 72.
- BRAVO, J. M. **Otimização da operação de um reservatório para controle de cheias com base na previsão de vazão**. Dissertação (Mestrado) — UFRGS, 2006. Citado na página 32.
- CARTER, M. **Foundations of Mathematical Economics (The MIT Press)**. [S.l.]: The MIT Press, 2001. ISBN 0262531925. Citado na página 10.
- COELLO, C. C.; LECHUGA, M. MOPSO: a proposal for multiple objective particle swarm optimization. In: **Proceedings of the 2002 Congress on Evolutionary Computation. CEC02 (Cat. No.02TH8600)**. [S.l.]: IEEE, 2002. Citado 3 vezes nas páginas 6, 11 e 26.
- DEB, K. **Multi-Objective Optimization Using Evolutionary Algorithms**. New York, NY, USA: John Wiley & Sons, Inc., 2001. ISBN 047187339X. Citado 6 vezes nas páginas 9, 10, 20, 21, 23 e 24.
- DEB, K. A fast and elitist multiobjective genetic algorithm: NSGA II. **Evolutionary Computation, IEEE Transactions on**, v. 6, 2002. Citado 4 vezes nas páginas 6, 11, 19 e 21.
- ETHZ. **Density and approximations of -distribution for different testproblems**. 2019. Disponível em: <<https://sop.tik.ee.ethz.ch/download/supplementary/testproblems/>>. Acesso em: 14 de fevereiro de 2019. Citado na página 41.
- FERNANDES, J. P. T. **Abordagem lexicográfica na otimização da operação de usinas hidrelétricas**. Tese (Doutorado) — UNICAMP, 2015. Citado 2 vezes nas páginas 4 e 30.
- FINARDI, E. **Alocação de unidades geradoras hidrelétricas em sistemas hidrotérmicos utilizando relaxação lagrangeana e programação quadrática seqüencial**. Tese (Doutorado) — UFSC, 2003. Citado na página 32.
- FRANK, S.; STEPONAVICE, I.; REBENNACK, S. Optimal power flow: a bibliographic survey II. **Energy Systems**, Springer Nature, v. 3, n. 3, p. 259–289, apr 2012. Citado na página 11.
- GIL, A. C. **Como Elaborar Projetos de Pesquisa**. [S.l.]: Atlas, 2010. ISBN 8522458235. Citado na página 36.
- GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization, and Machine Learning**. [S.l.]: Addison-Wesley Professional, 1989. ISBN 0201157675. Citado na página 28.

HENGGELE, C.; RICARDO, T.; ANTUNES, H. C. **Manual de Computação Evolutiva e Meta Heurística**. [S.l.]: UFMG, 2013. ISBN 8542300467. Citado na página 11.

HUANG, V.; QIN, A.; SUGANTHAN, P. Self-adaptive differential evolution algorithm for constrained real-parameter optimization. In: **2006 IEEE International Conference on Evolutionary Computation**. [S.l.]: IEEE, 2006. Citado na página 17.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: **Proceedings of ICNN95 - International Conference on Neural Networks**. [S.l.]: IEEE, 1995. Citado 3 vezes nas páginas 5, 11 e 14.

KORDON, A. **Applying Computational Intelligence: How to Create Value**. [S.l.]: Springer, 2009. ISBN 9783540699132. Citado na página 1.

LI, F.-F.; QIU, J. Multi-objective reservoir optimization balancing energy generation and firm power. **Energies**, MDPI AG, v. 8, n. 12, p. 6962–6976, jul 2015. Citado na página 4.

LISBOA, A.; GUEDES, L.; VIEIRA, D. Otimização multiobjetivo de usinas hidrelétricas em cascata. 2013. Citado 2 vezes nas páginas 4 e 55.

LOPES, H. S.; TAKAHASHI, R. H. C. **Computação Evolucionária em Problemas de Engenharia**. [S.l.]: Omnipax, 2011. ISBN 9788564619005. Citado na página 1.

MACIEL, R. S.; ROSA, M.; MIRANDA, V.; PADILHA-FELTRIN, A. Multi-objective evolutionary particle swarm optimization in the assessment of the impact of distributed generation. **Electric Power Systems Research**, Elsevier BV, v. 89, p. 100–108, aug 2012. Citado 5 vezes nas páginas 6, 11, 28, 29 e 47.

MARCELINO, C. **Otimização de um Sistema de Controle Conjunto de Geração de Energia Hidrelétrica com o emprego de Inteligência Computacional**. Dissertação (Mestrado) — CEFET-MG, 2012. Citado 6 vezes nas páginas 2, 3, 34, 36, 71 e 86.

MARCELINO, C. **Uma Abordagem Evolutiva e Híbrida para a Solução de Problemas de Fluxo de Potência Ótimo**. Tese (Doutorado) — CEFET-MG, 2017. Citado 10 vezes nas páginas 2, 3, 5, 7, 11, 33, 34, 36, 37 e 86.

MARCELINO, C.; WANNER, E.; ALMEIDA, P. A novel mathematical modeling approach to the electric dispatch problem: Case study using differential evolution algorithms. **Proceedings of conference: IEEE Congress on Evolutionary Computation (CEC)**, p. 400–407, 2013. Citado na página 2.

MARCELINO, C. G.; ALMEIDA, P.; WANNER, E. F.; BAUMANN, M.; WEIL, M.; CARVALHO, L.; MIRANDA, V. Solving security constrained optimal power flow problems: a hybrid evolutionary approach. **The International Journal of Research on Intelligent Systems for Real Life Complex Problems**, 2018. Citado na página 18.

MARCELINO, C. G.; ALMEIDA, P. E. M.; WANNER, E. F.; CARVALHO, L. M.; MIRANDA, V. Fundamentals of the c-DEEPSO algorithm and its application to the reactive power optimization of wind farms. In: **2016 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.]: IEEE, 2016. Citado na página 18.

MARCELINO, C. G.; ALMEIDA, P. E. M.; WANNER, E. F.; BAUMANN, M.; WEIL, M.; CARVALHO, L. M.; MIRANDA, V. Solving security constrained optimal power flow problems: a hybrid evolutionary approach. **Applied Intelligence**, Springer Nature America, Inc, v. 48, n. 10, p. 3672–3690, apr 2018. Citado 4 vezes nas páginas 3, 5, 17 e 36.

MARCELINO, C. G.; CARVALHO, L. M.; ALMEIDA, P. E. M.; WANNER, E. F.; MIRANDA, V. Application of Evolutionary Multiobjective Algorithms for solving the problem of Energy Dispatch in Hydroelectric Power Plants. In: **8th International Conference on Evolutionary Multi-Criterion Optimization**, 2015, p. 1–15, 2015. Citado 5 vezes nas páginas 3, 35, 37, 67 e 84.

MIRANDA, V.; ALVES, R. Differential evolutionary particle swarm optimization (DEEPSO): A successful hybrid. In: **2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence**. [S.l.]: IEEE, 2013. Citado 4 vezes nas páginas 5, 11, 16 e 17.

MIRANDA, V.; FONSECA, N. EPSO - best-of-two-worlds meta-heuristic applied to power system problems. In: **Proceedings of the 2002 Congress on Evolutionary Computation. CEC02 (Cat. No.02TH8600)**. [S.l.]: IEEE, 2002. Citado 3 vezes nas páginas 5, 11 e 15.

MONTGOMERY, D. C. **Estatística Aplicada a Engenharia**. LTC, 2004. ISBN 8521613989. Disponível em: <<https://www.amazon.com/Estatistica-Aplicada-Engenharia-Douglas-Montgomery/dp/8521613989?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=8521613989>>. Citado na página 38.

MONTGOMERY, D. C. **Design and Analysis of Experiments**. [S.l.]: Wiley, 2012. ISBN 1118146921. Citado 2 vezes nas páginas 39 e 40.

MOSTAGHIM, S.; TEICH, J. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In: **Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS03 (Cat. No.03EX706)**. [S.l.]: IEEE, 2003. Citado 3 vezes nas páginas 6, 27 e 47.

NGUYEN, T. T.; YANG, S.; BRANKE, J. Evolutionary dynamic optimization: A survey of the state of the art. **Swarm and Evolutionary Computation**, Elsevier BV, v. 6, p. 1–24, out. 2012. Citado na página 11.

ONS. 2018. Disponível em: <<http://www.ons.org.br>>. Acesso em: 20 de fevereiro de 2018. Citado na página 2.

PRICE, K.; STORN, R. M.; LAMPINEN, J. A. **Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)**. [S.l.]: Springer, 2006. ISBN 978-3-540-31306-9. Citado na página 13.

PRICE, K. V. New ideas in optimization. In: CORNE, D.; DORIGO, M.; GLOVER, F.; DAS-GUPTA, D.; MOSCATO, P.; POLI, R.; PRICE, K. V. (Ed.). Maidenhead, UK, England: McGraw-Hill Ltd., UK, 1999. cap. An Introduction to Differential Evolution, p. 79–108. ISBN 0-07-709506-5. Citado na página 13.

QIN, A.; SUGANTHAN, P. Self-adaptive differential evolution algorithm for numerical optimization. In: **2005 IEEE Congress on Evolutionary Computation**. [S.l.]: IEEE, 2005. Citado na página 17.

RIQUELME, N.; LÜCKEN, C. V.; BARAN, B. Performance metrics in multi-objective optimization. In: IEEE. **Computing Conference (CLEI), 2015 Latin American**. [S.l.], 2015. p. 1–11. Citado na página 38.

SCHREIBER, G. **Usinas Hidrelétricas**. [S.l.]: Rio de Janeiro, 1981. Citado na página 30.

SCUZZIATO, M. R. **Proposta de um modelo para alocação ótima de unidades hidrelétricas para usinas em cascata**. Dissertação (Mestrado) — UFSC, 2011. Citado na página 4.

SILVERMAN, B. W. **Density Estimation for Statistics and Data Analysis**. [S.l.]: Chapman and Hall, 1986. ISBN 0412246201. Citado na página 24.

STORN, R.; PRICE, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, Dec 1997. ISSN 1573-2916. Citado 4 vezes nas páginas 5, 11, 12 e 13.

VELDHUIZEN, D. A. V.; LAMONT, G. B. **Multiobjective Evolutionary Algorithm Research: A History and Analysis**. 1998. Citado na página 39.

XUE, F.; SANDERSON, A.; GRAVES, R. Pareto-based multi-objective differential evolution. In: **The 2003 Congress on Evolutionary Computation, 2003. CEC 03**. [S.l.]: IEEE, 2003. Citado 4 vezes nas páginas 6, 11, 27 e 28.

ZITZLER, E. Evolutionary algorithms for multiobjective optimization: Methods and applications. Citeseer, 1999. Citado na página 38.

ZITZLER, E.; DEB, K.; THIELE, L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. **Evolutionary Computation**, v. 8, n. 2, p. 173–195, 2000. Citado na página 41.

ZITZLER, E.; LAUMANN, M.; THIELE, L. **SPEA2: Improving the Strength Pareto Evolutionary Algorithm**. [S.l.], 2001. Citado 4 vezes nas páginas 6, 11, 23 e 25.

ZITZLER, E.; THIELE, L. Multiobjective optimization using evolutionary algorithms — a comparative case study. In: EIBEN, A. E.; BÄCK, T.; SCHOENAUER, M.; SCHWEFEL, H.-P. (Ed.). **Parallel Problem Solving from Nature — PPSN V**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. p. 292–301. ISBN 978-3-540-49672-4. Citado na página 39.