

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL

LUCAS MARCONDES PAVELSKI

**OTIMIZAÇÃO EVOLUTIVA MULTIOBJETIVO BASEADA EM
DECOMPOSIÇÃO E ASSISTIDA POR MÁQUINAS DE
APRENDIZADO EXTREMO**

DISSERTAÇÃO

CURITIBA

2015

LUCAS MARCONDES PAVELSKI

**OTIMIZAÇÃO EVOLUTIVA MULTIOBJETIVO BASEADA EM
DECOMPOSIÇÃO E ASSISTIDA POR MÁQUINAS DE
APRENDIZADO EXTREMO**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Mestre em Ciências” – Área de Concentração: Informática Industrial.

Orientadora: Myriam Regattieri Delgado

Coorientadora: Carolina Paula de Almeida

CURITIBA

2015

Dados Internacionais de Catalogação na Publicação

P337o Pavelski, Lucas Marcondes
2015 Otimização evolutiva multiobjetivo baseada em decomposição e assistida por máquinas de aprendizado extremo / Lucas Marcondes Pavelski.-- 2015.
89 f.: il.; 30 cm

Texto em português, com resumo em inglês.
Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Curitiba, 2015.
Bibliografia: f. 80-84

1. Algoritmos. 2. Computação evolutiva. 3. Otimização matemática. 4. Processo decisório por critério múltiplo. 5. Aprendizado do computador. 6. Método de decomposição. 7. Métodos de simulação. 8. Engenharia elétrica - Dissertações. I. Delgado, Myriam Regattieri De Biase da Silva, orient. II. Almeida, Carolina Paula de, coorient. III. Universidade Tecnológica Federal do Paraná - Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. IV. Título.

CDD 22 -- 621.3

Biblioteca Central da UTFPR, Câmpus Curitiba

Titulo da Dissertação Nº. _____

Otimização Evolutiva Multiobjetivo Baseada em Decomposição e Assistida por Máquinas de Aprendizado Extremo

Por

Lucas Marcondes Pavelski

Orientadora: Prof.^a Dra. Myriam Regattieri de Biase da Silva Delgado
Coorientadora: Prof.^a Dra. Carolina Paula De Almeida

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM CIÊNCIAS – Área de Concentração: Engenharia de Computação do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI – da Universidade Tecnológica Federal do Paraná – UTFPR, às 14h do dia 26 de fevereiro de 2015. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores doutores:

Prof.^a Dra. Myriam Regattieri de Biase da Silva
Delgado (Presidente – UTFPR)

Prof. Dr. Fernando José Von Zuben
(UNICAMP)

Prof. Dr. Gilberto Reynoso Meza
(PUC-PR)

Prof. Dr. Richard Aderbal Gonçalves
(UNICENTRO)

Visto da coordenação

Prof. Emilio Carlos Gomes Wille, Dr.
(Coordenador do CPGEI)

RESUMO

PAVELSKI, Lucas M.. OTIMIZAÇÃO EVOLUTIVA MULTIOBJETIVO BASEADA EM DECOMPOSIÇÃO E ASSISTIDA POR MÁQUINAS DE APRENDIZADO EXTREMO. 89 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2015.

Muitos problemas de otimização reais apresentam mais de uma função-objetivo. Quando os objetivos são conflitantes, estratégias especializadas são necessárias, como é o caso dos algoritmos evolutivos multiobjetivo (MOEAs, do inglês *Multi-objective Optimization Evolutionary Algorithms*). Entretanto, se a avaliação das funções-objetivo é custosa (alto custo computacional ou econômico) muitos MOEAs propostos são impraticáveis. Uma alternativa pode ser a utilização de um modelo de aprendizado de máquina que aproxima o cálculo do *fitness* (*surrogate*) no algoritmo de otimização. Este trabalho propõe e investiga uma plataforma chamada ELMOEA/D que agrega MOEAs do estado da arte baseados em decomposição de objetivos (MOEA/D) e máquinas de aprendizado extremo (ELMs, do inglês *Extreme Learning Machines*) como modelos *surrogate*. A plataforma proposta é testada com diferentes variantes do algoritmo MOEA/D e apresenta bons resultados em problemas *benchmark*, comparada a um algoritmo da literatura que também utiliza MOEA/D mas modelos *surrogates* baseados em redes com função de base radial. A plataforma ELMOEA/D também é testada no Problema de Predição de Estrutura de Proteínas (PPEP). Apesar dos resultados alcançados pela proposta não serem tão animadores quanto aqueles obtidos nos *benchmarks* (quando comparados os algoritmos com e sem *surrogates*), diversos aspectos da proposta e do problema são explorados. Por fim, a plataforma ELMOEA/D é aplicada a uma formulação alternativa do PPEP com sete objetivos e, com estes resultados, várias direções para trabalhos futuros são apontadas.

Palavras-chave: otimização multiobjetivo, decomposição, aproximação de *fitness*, máquinas de aprendizado extremo.

ABSTRACT

PAVELSKI, Lucas M.. MULTI-OBJECTIVE EVOLUTIONARY OPTIMIZATION BASED ON DECOMPOSITION ASSISTED BY EXTREME LEARNING MACHINES. 89 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2015.

Many real optimization problems have more than one objective function. When the objectives are in conflict, there is a need for specialized strategies, as is the case of the Multi-objective Optimization Evolutionary Algorithms (MOEAs). However, if the functions evaluation is expensive (high computational or economical costs) many proposed MOEAs are impractical. An alternative might be the use of a machine learning model to approximate the fitness function (surrogates) in the optimization algorithm. This work proposes and investigates a framework called ELMOEA/D that aggregates state-of-the-art MOEAs based on decomposition of objectives (MOEA/D) and extreme learning machines as surrogate models. The proposed framework is tested with different MOEA/D variants and show good results in benchmark problems, compared to a literature algorithm that also encompasses MOEA/D but uses surrogate models based on radial basis function networks. The ELMOEA/D framework is also applied to the protein structure prediction problem (PSPP). Despite the fact that the results achieved by the proposed approach were not as encouraging as the ones achieved in the benchmarks (when the algorithms with and without surrogates are compared), many aspects of both algorithm and problem are explored. Finally, the ELMOEA/D framework is applied to an alternative formulation of the PSPP and the results lead to various directions for future works.

Keywords: multi-objective optimization, decomposition, surrogate, extreme learning machines.

LISTA DE FIGURAS

FIGURA 1	– Mapeamento do espaço de decisão para o espaço objetivo; relação, conjunto e frente de Pareto	21
FIGURA 2	– <i>Bless and curse of uncertainty</i> em aproximações de modelos <i>surrogate</i> .	24
FIGURA 3	– Exemplo de soluções geradas pelos operadores DE	28
FIGURA 4	– ELM como uma rede neural de camada intermediária única e múltiplas saídas	35
FIGURA 5	– Funções de ativação	35
FIGURA 6	– Estrutura dos aminoácidos	37
FIGURA 7	– Ligação peptídea entre aminoácidos	38
FIGURA 8	– Representação de uma proteína em diferentes níveis estruturais	38
FIGURA 9	– Estrutura da proteína <i>Met-Enkephalin</i>	42
FIGURA 10	– Fluxo de execução da plataforma ELMOEA/D	47
FIGURA 11	– Associação entre os pesos do MOEA/D e os pesos utilizados na seleção	50
FIGURA 12	– Frente de Pareto dos problemas <i>benchmark</i> ZDT	57
FIGURA 13	– Exemplo de \mathcal{PF} encontradas pelas variantes do algoritmo ELMOEA/D utilizando MOEA/D original e MOEA/D-STM	62
FIGURA 14	– Função de <i>attainment</i> para o MOP ZDT1	62
FIGURA 15	– Função de <i>attainment</i> para o MOP ZDT6	63
FIGURA 16	– Resultados para diferentes variantes do otimizador MOEA/D no algoritmo ELMOEA/D	66
FIGURA 17	– Comparação do algoritmo ELMOEA/D e MOEA/D nas configurações iniciais	67
FIGURA 18	– Resultados para diferentes tamanhos de população e número de gerações do otimizador MOEA/D-DE no algoritmo ELMOEA/D	69
FIGURA 19	– Resultados para diferentes funções de agregação no algoritmo ELMOEA/D	70
FIGURA 20	– Resultados para diferentes operadores DE e hiper-heurísticas no algoritmo ELMOEA/D	71
FIGURA 21	– Resultados para múltiplas ELMs como <i>surrogates</i> no algoritmo ELMOEA/D	72
FIGURA 22	– Comparação dos algoritmos ELMOEA/D e MOEA/D-DE nas configurações finais	73
FIGURA 23	– Comparação dos algoritmos ELMOEA/D e MOEA/D ao longo do tempo	75
FIGURA 24	– Resultados para PPEP com sete objetivos no algoritmo ELMOEA/D ...	76

LISTA DE TABELAS

TABELA 1	– Número de ângulos livres de cada resíduo	41
TABELA 2	– Restrições dos ângulos diedrais (Φ e Ψ) para cada estrutura secundária .	41
TABELA 3	– Formulação dos problemas ZDT	57
TABELA 4	– Parâmetros dos <i>benchmarks</i> e ELM	58
TABELA 5	– Parâmetros ELMOEA/D para os testes com <i>benchmarks</i>	58
TABELA 6	– Comparação entre variantes do algoritmo MOEA/D na plataforma ELMOEA/D	60
TABELA 7	– Comparação do ELMOEA/D com algoritmos da literatura	64
TABELA 8	– Características das proteínas utilizadas nos testes com PPEP multiobjetivo	65
TABELA 9	– Resultados dos testes com diferentes parâmetros e configurações para o PPEP das Seções 4.3.2 e 4.3.4	68
TABELA 10	– Resultados do algoritmo ELMOEA/D para diferentes parâmetros do MOEA/D	86
TABELA 11	– Resultados do algoritmo ELMOEA/D para diferentes funções de agregação	87
TABELA 12	– Resultados do algoritmo ELMOEA/D para diferentes operadores de cruzamento	88
TABELA 13	– Resultados do algoritmo ELMOEA/D para múltiplos <i>surrogates</i>	89

LISTA DE ALGORITMOS

ALGORITMO 1	–	MOEA/D	25
ALGORITMO 2	–	MOEA/D-DE	27
ALGORITMO 3	–	SeleçãoSTM	30
ALGORITMO 4	–	MOEA/D-STM	31
ALGORITMO 5	–	ELMOEA/D	47

LISTA DE SIGLAS

DE	Evolução diferencial, do inglês <i>Differential Evolution</i> (ver Seção 2.3.1, Página 26)
DE/rand/1/bin	Operador DE utilizando pais aleatórios, 1 diferença entre pares e combinação binária com vetor alvo (ver Equação 8, Página 27)
DE/rand/2/bin	Operador DE utilizando pais aleatórios, 2 diferenças entre pares e combinação binária com vetor alvo (ver Seção 2.3.1, Página 26)
DE/nonlinear	Operador DE que combina indivíduos utilizando uma função quadrática (ver Equação 10, Página 28)
ELM	Máquina de aprendizado extremo, do inglês <i>Extreme Learning Machine</i> (ver Seção 2.5, Página 34)
ELMOEA/D	MOEA/D com <i>surrogates</i> ELM (ver Capítulo 3, Página 43)
FRRMAB	<i>Fitness-Rate-Rank-Based Multi-armed Bandit</i> (ver Seção 2.4, Página 32)
GAU	Função de ativação gaussiana (ver Equação 19, Página 35)
LHS	<i>Latin Hypercube Sampling</i> (MCKAY <i>et al.</i> , 1979)
MOEA	Algoritmo evolucionário multiobjetivo, do inglês <i>Multi-objective Optimization Evolutionary Algorithm</i> (ver Seção 2.1, Página 20)
MOEA/D	<i>Multi-objective Optimization Evolutionary Algorithm based on Decomposition</i> (ver Seção 2.3, Página 24)
MOEA/D-DE	<i>MOEA/D with Differential Evolution</i> (ver Seção 2.3.1, Página 26)
MOEA/D-DRA	<i>MOEA/D with Dynamic Resource Allocation</i> (ver Seção 2.3.2, Página 28)
MOEA/D-RBF	<i>MOEA/D assisted by Radial Basis Function surrogates</i> (MARTÍNEZ; COLLO, 2013)
MOEA/D-STM	<i>MOEA/D with Stable Matching</i> (ver Seção 2.3.2, Página 28)
MOP	problema de otimização multiobjetivo, do inglês <i>Multi-objective Optimization Problem</i> (ver Seção 2.1, Página 20)
MQ	Função de ativação <i>multiquadric</i> (ver Equação 20, Página 35)
NSGA-II	<i>Non-dominated Sorting Genetic Algorithm II</i> (DEB <i>et al.</i> , 2000)
NTCHE	Função de agregação Tchebycheff Normalizada (ver Equação 6, Página 25)
PBI	Função de agregação <i>Penalty Boundary Intersection</i> (ver Equação 7, Página 26)
PDB	<i>Protein Data Bank</i> (BERMAN <i>et al.</i> , 2000)
PM	<i>Probability Matching</i> (ver Seção 2.4, Página 32)
PSP	Predição de estrutura de proteínas, do inglês <i>Protein Structure Prediction</i> (ver Seção 2.6, Página 36)
SIG	Função de ativação sigmoideal (ver Equação 18, Página 34)
RBF	Função de base radial, do inglês <i>Radial Basis Function</i> (HAYKIN, 1998)
TCHE	Função de agregação Tchebycheff (ver Equação 5, Página 25)

LISTA DE SÍMBOLOS

Ω	Espaço de variáveis de decisão ou espaço de decisão (ver Figura 1, Página 21)
Y	Espaço das funções-objetivo ou espaço objetivo (ver Figura 1, Página 21)
N_D	Número de variáveis ou dimensão do espaço de decisão (ver Equação 1, Página 20)
N_O	Número de objetivos ou dimensão do espaço objetivo (ver Equação 1, Página 20)
\mathbf{x}_L	Vetor dos limites inferiores para cada variável (ver Equação 1, Página 20)
\mathbf{x}_U	Vetor dos limites superiores para cada variável (ver Equação 1, Página 20)
$\mathbf{x}^a \prec \mathbf{x}^b$	Relação de Pareto, \mathbf{x}^a domina \mathbf{x}^b (ver Equação 2, Página 21)
\mathcal{PS}	Conjunto de Pareto, do inglês <i>Pareto Set</i> (ver Equação 3, Página 22)
\mathcal{PF}	Frente de Pareto, do inglês <i>Pareto Front</i> (ver Equação 3, Página 22)
\mathcal{APS}	Aproximação do conjunto de Pareto (ver Seção 2.1, Página 20)
N	Tamanho da população no algoritmo MOEA/D (ver Seção 2.3, Página 24)
W	Conjunto de pesos no algoritmo MOEA/D (ver Seção 2.3, Página 24)
P	População no algoritmo MOEA/D (ver Seção 2.3, Página 24)
N_B	Número de vizinhos por indivíduo no algoritmo MOEA/D (ver Seção 2.3, Página 24)
$B(\mathbf{x})$	Vizinhos do indivíduo \mathbf{x} no algoritmo MOEA/D (ver Seção 2.3, Página 24)
$F(\mathbf{x} \mathbf{w}, \mathbf{z})$	Função de agregação para um indivíduo \mathbf{x} , dados o vetor de pesos \mathbf{w} e o ponto ideal \mathbf{z} , no algoritmo MOEA/D (ver Seção 2.3, Página 24)
N_R	Número máximo de substituições por indivíduo no algoritmo MOEA/D-DE (ver Seção 2.3.1, Página 26)
CR	Taxa de cruzamento no algoritmo DE, do inglês <i>Crossover Rate</i> (ver Seção 2.3.1, Página 26)
F	Fator de escalarização no algoritmo DE, do inglês <i>Scalling Factor</i> (ver Seção 2.3.1, Página 26)
N_{Op}	Número de operadores disponíveis nas estratégias de seleção adaptativa de operadores (ver Seção 2.4, Página 32)
r_k	Valor de recompensa para o k -ésimo operador nas estratégias de seleção adaptativa de operadores (ver Seção 2.4, Página 32)
p_{min}	Probabilidade mínima na seleção na seleção adaptativa de operadores PM (ver Seção 2.4, Página 32)
α	Taxa de aprendizado na seleção adaptativa de operadores PM (ver Seção 2.4, Página 32)
N_{sw}	Tamanho da janela deslizante na seleção adaptativa de operadores FFRMAB (ver Seção 2.4, Página 32)
C_s	Parâmetro de escalarização na seleção adaptativa de operadores FFRMAB (ver Seção 2.4, Página 32)
D	Parâmetro de decaimento na seleção adaptativa de operadores FFRMAB (ver Seção 2.4, Página 32)
$f_{ELM}(\mathbf{x})$	Função de saída na rede ELM (ver Seção 2.5, Página 34)
$h(\mathbf{x})$	Função de ativação para a entrada \mathbf{x} na rede ELM (ver Seção 2.5, Página 34)
N_H	Número de neurônios ocultos na rede ELM (ver Seção 2.5, Página 34)
C	Parâmetro regularizador na rede ELM (ver Seção 2.5, Página 34)

β	Vetor de pesos de saída na rede ELM (ver Seção 2.5, Página 34)
N_T	Número de entradas/saídas para treinamento na rede ELM (ver Seção 2.5, Página 34)
Φ_R e Ψ_R	Ângulos diedrais do resíduo R (ver Seção 2.6, Página 36)
$\chi_{k,R}$	k -ésimo ângulo livre do resíduo R (ver Seção 2.6, Página 36)
N_E	Número máximo de avaliações da função custosa f no algoritmo ELMOEA/D (ver Capítulo 3, Página 43)
P_E	Arquivo com as soluções avaliadas pela função custosa f no algoritmo ELMOEA/D (ver Capítulo 3, Página 43)
P_T	Arquivo com as soluções para treinamento dos <i>surrogates</i> no algoritmo ELMOEA/D (ver Capítulo 3, Página 43)
W_T	Conjunto de pesos para atualização do arquivo de treinamento (ver Capítulo 3, Página 43)
W_S	Conjunto de pesos para seleção de soluções a serem avaliadas com a função custosa f (ver Capítulo 3, Página 43)
N_S	Número de soluções selecionadas para avaliação com a função custosa f em cada iteração do algoritmo ELMOEA/D (ver Capítulo 3, Página 43)
ε	Distância mínima entre duas soluções no arquivo de soluções avaliadas (ver Capítulo 3, Página 43)
N_a	Número de <i>surrogates</i> no algoritmo ELMOEA/D com múltiplos <i>surrogates</i> (ver Seção 3.2.6, Página 52)
$I_H(P, \mathbf{r})$	Métrica Hipervolume para a população P e ponto de referência \mathbf{r} (ver Equação 35, Página 55)
$DM(P)$	Menor soma dos objetivos para os indivíduos em P (ver Equação 36, Página 55)

SUMÁRIO

1 INTRODUÇÃO	13
1.1 CARACTERIZAÇÃO DO PROBLEMA	15
1.2 MOTIVAÇÃO	16
1.3 OBJETIVOS	17
1.3.1 Objetivos Específicos	17
1.3.2 Contribuições	18
1.4 ESTRUTURA DO TRABALHO	19
2 FUNDAMENTAÇÃO	20
2.1 ALGORITMOS EVOLUTIVOS PARA OTIMIZAÇÃO MULTIOBJETIVO	20
2.2 ALGORITMOS EVOLUTIVOS PARA PROBLEMAS COM AVALIAÇÃO CUS- TOSA	23
2.3 MOEA/D E DECOMPOSIÇÃO DE OBJETIVOS	24
2.3.1 MOEA/D-DE e Evolução Diferencial	26
2.3.2 MOEA/D-STM	28
2.3.3 MOEA/D-RBF	31
2.4 SELEÇÃO ADAPTATIVA DE OPERADORES	32
2.5 <i>EXTREME LEARNING MACHINE</i> (ELM)	34
2.6 PROBLEMA DA PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS (PPEP)	36
2.6.1 Campo de força CHARMM	39
2.6.2 Formulação do PPEP multiobjetivo	40
3 MOEA/D COM SURROGATES ELM (ELMOEA/D)	43
3.1 TRABALHOS CORRELATOS	43
3.1.1 Modelos <i>surrogates</i> em MOEAs	43
3.1.2 Modelos <i>surrogates</i> com o algoritmo MOEA/D	45
3.1.3 ELM como modelo <i>surrogate</i>	45
3.2 ELMOEA/D: ALGORITMO MULTIOBJETIVO BASEADO EM DECOMPOSIÇÃO E SURROGATES ELM	46
3.2.1 Geração de soluções iniciais	47
3.2.2 Treinamento do <i>surrogate</i> ELM	48
3.2.3 População inicial e otimização da função aproximada com MOEA/D	49
3.2.4 Seleção de soluções para avaliação	50
3.2.5 Atualização dos arquivos	51
3.2.6 Múltiplos <i>surrogates</i>	52
3.3 DIFERENÇAS COM O ALGORITMO MOEA/D-RBF	53
4 EXPERIMENTOS E RESULTADOS	54
4.1 METODOLOGIA DE AVALIAÇÃO E COMPARAÇÃO	55
4.2 PARTE I: PROBLEMAS <i>BENCHMARK</i>	56
4.3 PARTE II: PPEP MULTIOBJETIVO	64
4.3.1 Melhor variante e comparação com MOEA/D-DE sem <i>surrogate</i>	65
4.3.2 Ajuste de parâmetros	68
4.3.3 Comparação final com MOEA/D-DE sem <i>surrogates</i>	72

4.3.4 PPEP com sete objetivos	74
5 CONCLUSÕES	77
REFERÊNCIAS	80
Apêndice A – RESULTADOS PARA DIFERENTES PARÂMETROS DO ELMOEA/D	
EM PROBLEMAS <i>BENCHMARK</i>	85

1 INTRODUÇÃO

Grande parte dos processos com os quais nos deparamos estão associados à otimização. Desde pessoas que procuram executar tarefas com mais eficiência até a própria natureza que faz com que sistemas físicos tendam a um estado de energia mínima. O estudo da otimização é, portanto, importante na tomada de decisão e análise de sistemas físicos.

O primeiro passo para otimizar um processo é a modelagem do problema, a qual envolve (NOCEDAL; WRIGHT, 2006): (1) as *variáveis* ou *variáveis de decisão*, que controlam as características do sistema e podem assumir diferentes valores, alterando diretamente o estado do sistema, por exemplo: a quantia investida em dado negócio ou as coordenadas da posição de cada átomo em certa molécula de um composto; (2) o *objetivo*, que é a medida quantitativa da qualidade do sistema em dado estado. Pode ser representado por uma função que retorna um número real e deve ser minimizado ou maximizado, como o lucro obtido sobre um investimento ou a energia total do sistema; e (3) as *restrições*, que impõem limitações ou penalidades quanto ao uso de recursos, por exemplo, investimentos não devem exceder determinado orçamento ou os limites da região onde certos átomos tendem a se posicionar. Dadas as restrições, o conjunto de todos os valores possíveis que as variáveis podem assumir sem violar nenhuma restrição é chamado de *conjunto factível*.

Com estas informações, o problema pode ser formulado e resolvido com o auxílio de computadores e algoritmos de otimização. De acordo com o Teorema da Inexistência do Almoço Grátis (WOLPERT; MACREADY, 1997), não existe um único algoritmo para resolver qualquer problema, e sim, um conjunto de técnicas especializadas para diferentes tipos de problema. Dentre os vários algoritmos que podem ser empregados para otimização, os algoritmos evolutivos destacam-se por sua flexibilidade e bons resultados em vários domínios (ENGELBRECHT, 2002).

Um problema de otimização multiobjetivo (MOP, do inglês *Multi-objective Optimization Problem*) pode ser descrito como contendo: um vetor de variáveis de decisão, satisfazendo as restrições dadas, e um vetor de funções-objetivo a serem otimizadas. As funções-objetivo

geralmente estão em conflito entre si, deste modo “otimizar” significa encontrar a solução com valores aceitáveis de acordo com o tomador de decisão para cada objetivo (compromisso). Uma abordagem mais genérica para resolver um MOP é encontrar um conjunto de boas soluções, explicitando o compromisso (*trade-off*) entre os objetivos. Então dar ao tomador de decisão o conjunto e a escolha da melhor solução utilizando informações de nível mais elevado (experiência, gosto pessoal, estética, etc). Desse modo, a busca pelas soluções eficientes não-dominadas independe do problema (COELLO *et al.*, 2007).

O uso de algoritmos evolutivos para otimização multiobjetivo é motivado pelo fato destes utilizarem populações e explorarem as similaridades das soluções. O estudo de algoritmos evolutivos multiobjetivo (MOEAs, do inglês *Multi-objective Optimization Evolutionary Algorithms*) estabelece um importante campo de pesquisa e aplicações em vários domínios. Algumas questões em aberto são: a aplicação de MOEAs em problemas com muitos objetivos (mais de 3 funções), a comparação de MOEAs e influência de alguns indicadores de qualidade, resultados teóricos sobre o papel e dinâmica da população, entre outros (COELLO *et al.*, 2007) e problemas multiobjetivos custosos (JIN, 2011). Este trabalho em particular utiliza um MOEA do estado da arte, baseado em decomposição de objetivos (MOEA/D) e investiga empiricamente seu desempenho em uma plataforma para otimização de problemas custosos, inclusive em um MOP com muitos objetivos.

Em certos problemas, o uso de algoritmos evolutivos tradicionais pode exigir um número grande de avaliações da função-objetivo. Este fato torna-se um impedimento em aplicações em que: a avaliação é custosa em relação a algum recurso (tempo computacional, financeiro etc.) e/ou não há formulação analítica para o objetivo a ser otimizado (a qualidade da solução deve ser determinada por uma pessoa ou equipamento físico). Problemas com estas características não são raros em áreas onde são necessárias simulações de sistemas complexos, como na engenharia e bioinformática. Na literatura, problemas com avaliação custosa são tipicamente resolvidos fazendo uso de formulações simplificadas do problema (simulações mais simples, de casos restritos) ou *surrogates* (JIN, 2005).

Algoritmos baseados em *surrogates* utilizam modelos computacionais eficientes para aproximar a avaliação custosa do problema e com isso diminuir o número total de avaliações. Além disso, em certos casos, a utilização de modelos *surrogate* mais simples em problemas de busca multimodais podem amenizar “rugosidades” no espaço de busca. Existem várias propostas para estratégias de gerenciamento de modelos *surrogate* em algoritmos evolutivos, sendo que uma revisão abrangente pode ser encontrada em Jin (2005, 2011).

Vários MOEAs são propostos para resolver problemas com funções-objetivo custosas,

explorando modelos *surrogate* baseados em processos gaussianos, polinômios, redes neurais, redes de função de base radial, entre outros. Porém, ainda poucos trabalhos sobre algoritmos assistidos por *surrogates* são relacionados a máquinas de aprendizado extremo (ELMs, do inglês *Extreme Learning Machines*). ELMs podem ser vistas como redes neurais de camada única, onde o treinamento é realizado em um passo, minimizando o erro quadrático e a norma dos pesos de saída. Desde sua proposta, resultados da literatura mostram o bom desempenho de ELMs para classificação e regressão, ainda com um custo computacional reduzido e poucos parâmetros a serem especificados (HUANG *et al.*, 2004, 2012).

1.1 CARACTERIZAÇÃO DO PROBLEMA

Este trabalho aborda principalmente: a resolução de problemas de otimização contínuos, irrestritos (ou *bound constrained*) e multiobjetivo com o uso de algoritmos evolutivos e modelos *surrogate*. Uma plataforma que agrega MOEAs baseados em decomposição e ELMs é proposta, e testada em duas classes de MOPs *benchmarks* e o problema real de Predição de Estruturas de Proteínas. Espera-se contribuir com resultados empíricos referentes ao estudo de MOEAs assistidos por *surrogates*, analisando algoritmos do estado da arte para otimização de problemas contínuos multiobjetivo com um número limitado de avaliações da função-objetivo.

Os problemas *benchmark* ZDT (ZITZLER; THIELE, 1998), utilizados neste trabalho, possuem dois objetivos e possuem diferentes características importantes para otimização multi-objetivo (como formato da fronteira de Pareto, descontinuidades, fronteiras sub-ótimas, entre outras). O mesmo conjunto de problemas já foi útil para comparar diversos MOEAs (DEB *et al.*, 2000; ZITZLER *et al.*, 2002; ZITZLER; KÜNZLI, 2004; ZHANG; LI, 2007) e são amplamente investigados na literatura. A resolução destes MOPs com poucas avaliações de função e a capacidade de especificar o número de variáveis nos é interessante para o propósito de testar um MOEA baseado em *surrogate* e sua escalabilidade quanto à dimensão do problema.

Um problema bastante conhecido na área de bioinformática, em que a avaliação da função-objetivo pode demandar muitos recursos, é o Problema da Predição de Estruturas de Proteínas (PPEP). O PPEP consiste em encontrar a estrutura tridimensional de uma proteína, dada sua sequência de aminoácidos. Devido ao grande número de proteínas sequenciadas e à dificuldade de obter sua conformação tridimensional em laboratório, vários métodos computacionais são propostos. Do ponto de vista de aplicação, o PPEP é de extrema importância para a biologia e campos relacionados, visto que grande parte das funções executadas por proteínas são explicadas por sua estrutura tridimensional (TRAMONTANO, 2006). Cutello *et al.* (2006) propõem a formulação do PPEP como um MOP, em um método *ab initio* (não dependente de

informações sobre a estrutura de proteínas parecidas), que minimiza simultaneamente as energias dos termos ligados e não ligados.

1.2 MOTIVAÇÃO

Problemas com formulações contínuas surgem naturalmente nas ciências exatas e engenharias, e recentemente vêm surgindo nas ciências econômicas e de gestão (NOCEDAL; WRIGHT, 2006). Certos casos ainda exigem que mais de um objetivo seja otimizado e, quando os objetivos são conflitantes e nenhuma informação é dada *a priori*, é desejável que várias soluções correspondentes ao compromisso entre os objetivos sejam encontradas. Como mencionado, esta estratégia é mais genérica e possibilita ao tomador de decisão escolher a melhor solução, entender melhor o problema e a relação entre os objetivos.

Vários exemplos mostram o sucesso dos MOEAs, tornando-os os principais métodos para resolver MOPs (DEB, 2011), inclusive MOPs custosos, envolvendo modelos *surrogates* (JIN, 2011). Em particular, este trabalho faz uso do algoritmo MOEA/D, que recentemente obteve sucesso em competições envolvendo a resolução de MOPs com alto grau de complexidade (ZHANG; SUGANTHAN, 2009). Além disso, o MOEA/D é baseado em decomposição de objetivos, uma abordagem promissora presente em outros algoritmos do estado da arte de otimização multiobjetivo, inclusive para otimização com muitos objetivos (MOPs com mais de 3 objetivos), como o NSGA-III (DEB; JAIN, 2014).

MOPs com alto custo computacional associado ao cálculo do *fitness* e número médio de variáveis ainda não são explorados amplamente na literatura de algoritmos evolutivos baseados em *surrogates*. Algumas propostas existentes para resolver MOPs custosos de dimensões menores (2 até 30 variáveis) utilizam modelos *surrogate* computacionalmente caros e têm seu desempenho deteriorado em relação ao número de variáveis (MARTÍNEZ; COELLO, 2013; JIN, 2005, 2011). Ainda, no contexto de otimização multiobjetivo assistida por *surrogates*, nenhum trabalho do conhecimento dos autores abordou a questão de escalabilidade do custo computacional quanto ao número de objetivos. Este trabalho procura abranger problemas de 8 até 275 variáveis e mostra que a investigação do algoritmo proposto em problemas com muitos objetivos é possível mas ainda demanda uma investigação mais ampla em trabalhos futuros.

O algoritmo proposto é inspirado no MOEA/D-RBF (MARTÍNEZ; COELLO, 2013) que utiliza MOEA/D e redes neurais RBF (HAYKIN, 2008) para resolver problemas multiobjetivo custosos, devido aos seus bons resultados para um número médio de variáveis. Porém, neste trabalho, redes neurais do tipo ELM (HUANG *et al.*, 2004) foram empregadas como *surrogates*.

A análise de ELMs nesse contexto é interessante por de ser pouco explorada na literatura de *surrogates*. Pelo conhecimento dos autores, apenas Lim *et al.* (2007) e Hao e Liu (2014) utilizam ELMs em algoritmos de otimização mono-objetivo. Além disso, vários trabalhos mostram que ELMs possuem baixo custo computacional e poucos parâmetros a serem especificados (HUANG *et al.*, 2012, 2015).

Do ponto de vista de aplicação, o PPEP é um problema desafiador na área de bioinformática. Proteínas mediam a maioria das funções de um organismo, e todas estas são, em geral, determinadas por suas estruturas tridimensionais. Anfinsen *et al.* (1961), aclamados pelo Prêmio Nobel em 1972, provam que a estrutura tridimensional de uma proteína pode ser obtida a partir da sequência de aminoácidos que a compõem. Tendo em vista a grande quantidade de proteínas sequenciadas, quando comparada com a quantidade de proteínas onde a estrutura foi determinada, e a dificuldade de obter a sua conformação experimentalmente, os métodos computacionais de predição de estrutura de proteínas possuem imenso interesse desde a biologia até a medicina. Neste trabalho, o algoritmo proposto é aplicado ao Problema de Predição de Estrutura de Proteínas (PPEP) modelado como um problema multiobjetivo, onde a minimização de duas energias conflitantes (dos termos ligados e não-ligados) é mais conveniente, em uma abordagem *ab initio*, na qual o conhecimento prévio da estrutura de outras proteínas semelhantes não é necessário.

1.3 OBJETIVOS

Esta dissertação pretende, de forma geral, propor e investigar a plataforma ELMOEA/D, que agrega redes neurais tipo ELM como modelos *surrogate* e MOEAs baseados em decomposição, e aplicá-la em MOPs *benchmarks* com diversas propriedades, bem como ao problema PPEP multiobjetivo.

1.3.1 OBJETIVOS ESPECÍFICOS

- Fundamentar e revisar a literatura pertinente aos conceitos necessários de otimização multiobjetivo, MOEA/D e suas variantes, operadores de evolução diferencial, seleção adaptativa de operadores e ELMs para regressão. Bem como revisar trabalhos relacionados envolvendo MOEAs assistidos por *surrogates*, que também fazem uso do MOEA/D além das redes tipo ELM;
- implementar a plataforma ELMOEA/D, inspirada no algoritmo MOEA/D-RBF, a qual envolve o gerenciamento do algoritmo MOEA/D assistido por redes tipo ELM;

- testar diferentes variantes do algoritmo MOEA/D na plataforma ELMOEA/D em problemas *benchmark* e comparar com o algoritmo MOEA/D-RBF através da métrica Hipervolume;
- ajustar e aplicar a proposta ao problema PPEP, avaliar os resultados conforme o valor de energia mínima alcançado em sete proteínas diferentes e comparar a proposta com o algoritmo MOEA/D-DE sem auxílio de *surrogates*;
- aplicar a proposta a uma formulação alternativa do PPEP, com sete objetivos.

1.3.2 CONTRIBUIÇÕES

As principais contribuições do trabalho são:

Proposta da plataforma ELMOEA/D: como mencionado, uma plataforma para otimização multiobjetivo custosa que agrega o algoritmo MOEA/D, com *surrogates* tipo ELM, é implementada;

Proposta do uso de múltiplas redes ELM: a proposta do ELMOEA/D é estendida para fazer uso de múltiplos *surrogates*, o que pode levar a melhores aproximações e resultados mais robustos (JIN, 2005);

Investigação empírica de redes ELM como modelos *surrogate*: poucos trabalhos exploram redes tipo ELM como *surrogate*. As principais vantagens das ELMs verificadas neste trabalho são o melhor desempenho e a redução de parâmetros, comparada a outros modelos da literatura (processos Gaussianos, redes RBF, etc);

Aplicação de MOEAs assistidos por *surrogates* a MOPs com muitas variáveis: a resolução de problemas custosos com muitas variáveis ainda é um problema em aberto (JIN, 2005, 2011) e este trabalho traz resultados empíricos da aplicação do ELMOEA/D em MOPs de 8 até 275 variáveis;

Estudo de diferentes variantes do algoritmo MOEA/D com *surrogates*: os principais trabalhos envolvendo MOEA/D com *surrogates* (ZHANG *et al.*, 2010; MARTÍNEZ; COELLO, 2013) focam em duas variantes (MOEA/D-DE e MOEA/D original) e a plataforma ELMOEA/D é testada com três variantes (MOEA/D original, MOEA/D-DE e MOEA/D-STM), além de testar três diferentes operadores (DE/*rand/1/bin*, DE/*rand/2/bin* e DE/*nonlinear*) e duas estratégias de seleção adaptativa de operadores (PM e FRRMAB);

Aplicação de MOEAs assistidos por *surrogates* ao PPEP em poucas avaliações: exceto algumas observações prévias feitas por Neumaier (1997), a resolução do problema PPEP multiobjetivo com auxílio de *surrogates* é pouco explorada na literatura. Neste trabalho, a proposta ELMOEA/D é aplicada ao PPEP e algumas propriedades sobre a influência dos parâmetros são investigadas;

PPEP com sete objetivos: a utilização de sete tipos de energias como objetivos individuais no PPEP é inovadora. A melhor configuração do algoritmo proposto para o PPEP com dois objetivos é aplicada a essa formulação alternativa e, apesar dos resultados não apresentarem vantagem, a contribuição traz várias possibilidades para trabalhos futuros;

Contribuições para literatura: os resultados prévios da plataforma ELMOEA/D com problemas *benchmark* foram divulgadas no congresso BRACIS 2014 (PAVELSKI *et al.*, 2014) e a pesquisa obteve um convite para publicação de um artigo expandido na revista *Neurocomputing* em 2015.

1.4 ESTRUTURA DO TRABALHO

O trabalho está estruturado da seguinte maneira. Primeiramente, na Seção 2, conceitos pertinentes são fundamentados, como a otimização multiobjetivo, MOEA/D e suas variantes, operadores da evolução diferencial, seleção adaptativa de operadores e regressão por meio de ELMs. Na Seção 3, alguns trabalhos relevantes da literatura sobre MOEAs assistidos por *surrogates* são comentados para contextualização da proposta; a qual é descrita com a discussão dos passos da plataforma ELMOEA/D e suas diferenças com o algoritmo MOEA/D-RBF. Em seguida, na Seção 4, os resultados alcançados são descritos em duas fases: (1) testes da plataforma ELMOEA/D com diferentes variantes do MOEA/D e comparação com o algoritmo MOEA/D-RBF em *benchmarks*; (2) adaptação e aplicação da proposta no problema PPEP com análises dos resultados no ajuste de parâmetros. Finalmente, a Seção 5 conclui com a revisão do conteúdo apresentado e direções para trabalhos futuros.

2 FUNDAMENTAÇÃO

Este capítulo apresenta uma breve fundamentação dos conceitos tratados na proposta, incluindo otimização multiobjetivo, MOEAs baseados em decomposição de objetivos, seleção adaptativa de operadores e redes ELM para regressão.

2.1 ALGORITMOS EVOLUTIVOS PARA OTIMIZAÇÃO MULTIOBJETIVO

Algumas classificações para problemas de otimização pertinentes são (NOCEDAL; WRIGHT, 2006; COELLO *et al.*, 2007):

- um problema é dito *discreto* se seu conjunto factível é finito ou infinito enumerável (variáveis binárias ou inteiras); ou *contínuo* caso o conjunto factível seja infinito (variáveis reais ou complexas);
- problemas *com restrições*, onde as condições sobre os valores que as variáveis podem assumir são ditadas por funções de restrição, ou problemas *irrestritos*, caso as únicas restrições, se houver, sejam os limites de valores máximos e mínimos impostos às variáveis;
- problemas *mono-objetivo* quando possuem apenas uma função-objetivo (considera uma qualidade ou combinação de qualidades a serem otimizadas) ou *multiobjetivo* se houver mais de uma função a ser otimizada.

Um problema de otimização multiobjetivo (MOP) irrestrito pode ser formulado como (COELLO *et al.*, 2007; DEB, 2011):

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimizar}} \quad & \mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{N_O}(\mathbf{x})\} \\ \text{sujeito a} \quad & \mathbf{x} \in \Omega \text{ ou seja, } x_{L,d} \leq x_d \leq x_{U,d} \quad d = 1, 2, \dots, N_D \end{aligned} \quad (1)$$

onde: \mathbf{x} é um vetor de N_D variáveis de decisão a serem otimizadas. O conjunto Ω representa o espaço das variáveis de decisão (ou espaço de decisão) com os valores contidos nos limites inferiores (\mathbf{x}_L) e superiores (\mathbf{x}_U) do problema formulado. \mathbf{f} é um vetor com N_O funções-objetivo,

da forma $\mathbf{f} : \Omega \rightarrow Y$. O conjunto imagem Y é o espaço das funções-objetivo (ou espaço objetivo), que contém os pontos resultantes da avaliação dos elementos de Ω para cada objetivo (eixo do espaço objetivo) e geralmente é um subconjunto de \mathbb{R}^{N_O} . O mapeamento de \mathbf{f} nos dois espaços para o caso $N_D = 2, N_O = 2$ é mostrado na Figura 1.

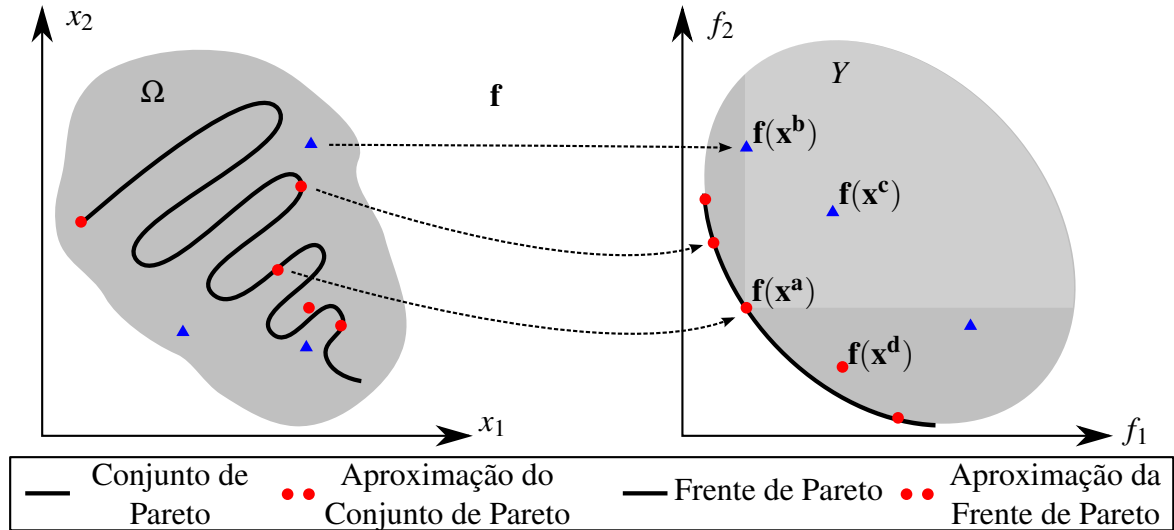


Figura 1: Mapeamento do espaço de decisão Ω para o espaço objetivo Y . Exemplos de dominância de Pareto: $\mathbf{x}^a \prec \mathbf{x}^b$ (apesar de serem equivalentes no primeiro objetivo, \mathbf{x}^a é melhor no segundo objetivo), $\mathbf{x}^a \prec \mathbf{x}^c$ (\mathbf{x}^a é melhor em ambos os objetivos) e $\mathbf{x}^a \not\prec \mathbf{x}^d$ (\mathbf{x}^a é melhor que \mathbf{x}^d no primeiro objetivo, porém é pior no segundo nesse caso \mathbf{x}^a e \mathbf{x}^d são incomparáveis em relação à dominância de Pareto).

Outro conceito importante e muito utilizado em otimização multiobjetivo é a relação de Pareto. Em um problema mono-objetivo, as qualidades das soluções estão diretamente relacionadas com os valores da função-objetivo e servem para comparar e ordenar as soluções. Porém, a tarefa de comparar duas soluções em um MOP não é trivial. Para vários objetivos, considera-se que uma solução \mathbf{x}^a *domina* \mathbf{x}^b ($\mathbf{x}^a \prec \mathbf{x}^b$) se \mathbf{x}^a é igual e, em ao menos um objetivo, melhor que \mathbf{x}^b . A relação de Pareto para um problema de minimização pode ser expressa matematicamente como (COELLO *et al.*, 2007):

$$\mathbf{x}^a \prec \mathbf{x}^b \rightarrow \forall j \in \{1, \dots, N_O\}, f_j(\mathbf{x}^a) \leq f_j(\mathbf{x}^b) \wedge \exists k \in \{1, \dots, N_O\} : f_k(\mathbf{x}^a) < f_k(\mathbf{x}^b) \quad (2)$$

A mesma terminologia será utilizada para se referir a um ponto que domina outro no espaço de objetivos. A Figura 1 mostra alguns exemplos da dominância de Pareto no espaço de objetivos (pontos vermelhos dominam pontos azuis).

A partir da relação de dominância de Pareto, também define-se o conjunto e frente de Pareto, bem como suas aproximações. O conjunto de Pareto (\mathcal{PS}) é formado pelas soluções que não podem ser melhoradas para um objetivo sem deteriorar outro, ou seja, soluções não-

dominadas por qualquer outra no conjunto Ω . A imagem do conjunto de Pareto no espaço objetivo é chamada de frente de Pareto (\mathcal{PF}) e suas formulações são (COELLO *et al.*, 2007):

$$\begin{aligned}\mathcal{PS} &:= \{ \mathbf{x} \in \Omega : \forall \mathbf{x}' \in \Omega, \mathbf{x}' \not\prec \mathbf{x} \} \\ \mathcal{PF} &:= \{ \mathbf{f}(\mathbf{x}) \in Y : \mathbf{x} \in \mathcal{PS} \}\end{aligned}\tag{3}$$

O conjunto e frente de Pareto podem conter um número infinito de elementos. Deste modo, algoritmos para otimização multiobjetivo geralmente buscam por uma aproximação do conjunto de Pareto (\mathcal{APS}) e a aproximação da frente de Pareto correspondente, formada por um número finito de soluções não dominadas (DEB, 2011; COELLO *et al.*, 2007)¹. A Figura 1 ilustra o conjunto e frente de Pareto, bem como suas aproximações.

O Tomador de Decisão, que possui informações de mais alto nível sobre o problema, pode ser utilizado para realizar o passo fundamental da escolha da solução que resolve o problema. A abordagem utilizada pode ser: (1) *a priori*, onde informações como preferências são especificadas antes da busca; (2) progressiva, em que o Tomador de Decisões é requisitado durante a busca e (3) *a posteriori*, para quando todas as soluções da aproximação da \mathcal{PF} encontrada são retornadas e a melhor é escolhida (COELLO *et al.*, 2007). Para estratégias genéricas, a abordagem *a posteriori* da tomada de decisão é preferível, pois não assume informações adicionais sobre o problema. Sendo assim, o algoritmo deve fornecer uma boa aproximação da \mathcal{PF} . Alguns critérios qualitativos de uma boa aproximação, segundo Zitzler (1999), são:

- a distância da aproximação para a \mathcal{PF} deve ser minimizada, para uma boa convergência;
- uma distribuição uniforme das soluções encontradas no espaço de objetivos é desejável, para uma boa cobertura;
- o espalhamento da aproximação obtida deve ser maximizado, ou seja, para cada objetivo, um grande intervalo de valores deve ser coberto, para uma boa diversidade.

Observa-se que o indicador Hipervolume representa a área ou volume dominado pela aproximação e serve como uma medida quantitativa que atente a todos os critérios de uma boa aproximação (ZITZLER; KÜNZLI, 2004). Com isso o Hipervolume é amplamente utilizado na literatura, bem como neste trabalho.

A ideia principal de um algoritmo evolutivo é tratar as diferentes soluções como indivíduos e explorar o espaço de busca como uma população que evolui. Os indivíduos possuem

¹Coello *et al.* (2007) define o conjunto de soluções não-dominadas encontradas por um MOEA como P_{known} , porém a tradução literal “Frente/Conjunto de Pareto conhecidos” causaria confusão caso a Frente/Conjunto de Pareto definitivos do problema sejam conhecidos.

uma medida de adequação ou *fitness*, cujo cálculo é baseado na função-objetivo, e interagem através de operadores, como cruzamento e mutação, gerando novos indivíduos. Os indivíduos mais adaptados são selecionados para formar a próxima geração e a população tenderá à solução ótima ao longo da busca (ENGELBRECHT, 2002). Algoritmos evolutivos mantêm populações de soluções e exploram suas similaridades, assim obtêm bons resultados em vários problemas de otimização mono-objetivo. Estas características tornam os algoritmos evolutivos interessantes para otimização multiobjetivo e incentivaram a criação e estudo dos MOEAs. Em muitos MOEAs, a população de soluções P pode ser representada como um conjunto de duplas formadas por uma solução \mathbf{x} e seus objetivos $\mathbf{f}(\mathbf{x})$:

$$P := \{ \langle \mathbf{x}_1, \mathbf{f}(\mathbf{x}_1) \rangle, \dots, \langle \mathbf{x}_N, \mathbf{f}(\mathbf{x}_N) \rangle \}.$$

Estudos e aplicações de MOEAs vêm crescendo em vários campos de pesquisa nas duas últimas décadas (DEB, 2011). Alguns exemplos de MOEAs mais referenciados são o *Pareto Archived Evolution Strategy* (PAES) (KNOWLES; CORNE, 1999), o *Non-Dominated Sorting Genetic Algorithm II* (NSGA-II) (DEB *et al.*, 2000), o *Strength Pareto Evolutionary Algorithm 2* (SPEA2) (ZITZLER *et al.*, 2002) e o *Indicator-based Evolutionary Algorithm* (IBEA) (ZITZLER; KÜNZLI, 2004). Recentemente, novas propostas de MOEAs têm utilizado a estratégia de decomposição de objetivos, como é o caso do algoritmo utilizado neste trabalho, o MOEA/D (ZHANG; LI, 2007).

2.2 ALGORITMOS EVOLUTIVOS PARA PROBLEMAS COM AVALIAÇÃO CUSTOSA

Com o objetivo de diminuir o número de avaliações em problemas custosos, os algoritmos evolutivos podem fazer uso de modelos computacionais eficientes (*surrogates*) para estimar o valor de *fitness*. Existem várias propostas para algoritmos evolutivos, incluindo MOEAs, assistidos por *surrogates*, explorando diferentes modos de gerenciamento (treinamento, adaptação) dos modelos. Ainda, a utilização de modelos *surrogate* não se limita a estimar o *fitness* durante a avaliação dos indivíduos, mas também na inicialização da população, entre gerações, na seleção de pais, no cruzamento, na mutação e na seleção de indivíduos (JIN, 2011).

Como o modelo *surrogate*, em muitos casos, pode ser menos complexo que a função original, outra capacidade dos algoritmos baseados em *surrogates* é a possibilidade de “suavizar” funções-objetivo multimodais que possuem “rugosidades” no espaço de busca. Ao mesmo tempo que, se não controlada, a aproximação gerada pelo modelo *surrogate* pode deteriorar a busca. Estes fenômenos são conhecidos como *bless and curse of uncertainty* (LIM *et al.*, 2010) e exemplificados na Figura 2.

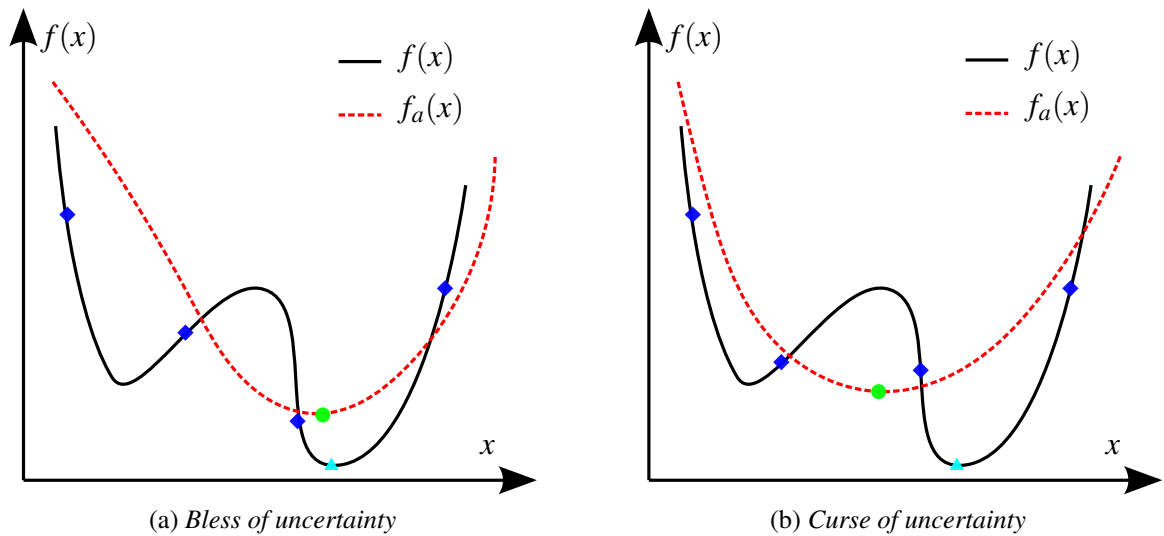


Figura 2: *Bless and curse of uncertainty* em aproximações de modelos *surrogate*: dada uma função-objetivo ($f(x)$) e sua aproximação por modelos *surrogate* ($f_a(x)$), baseada nos pontos conhecidos (\blacklozenge). No primeiro caso (a), o ponto ótimo da aproximação (\bullet) é próximo ao ponto ótimo da função original (\blacktriangle) e a utilização do modelo foi vantajosa. No segundo caso (b), a função aproximada não é vantajosa pois leva a acreditar que um ponto longe do ótimo da função serve para minimizá-la.

2.3 MOEA/D E DECOMPOSIÇÃO DE OBJETIVOS

O algoritmo MOEA/D faz uso de abordagens de agregação convencionais para decompor um MOP em vários problemas mono-objetivo (ZHANG; LI, 2007). Dado um conjunto $W = \{\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_N\}$ de N vetores de pesos, distribuídos uniformemente no espaço de objetivos, cada vetor de peso é associado a uma solução da população P . Relações de vizinhança são definidas de forma que uma solução \mathbf{x}_i associada ao vetor de pesos \mathbf{w}_i possui N_B vizinhos no conjunto $B(\mathbf{x}_i) = \{\mathbf{x}_{n1}, \mathbf{x}_{n2}, \dots, \mathbf{x}_{nN_B}\}$, correspondente às soluções cujos vetores de pesos são os N_B mais próximos a \mathbf{w}_i , de acordo com a distância euclidiana.

O Algoritmo 1 mostra o procedimento utilizado pelo MOEA/D para otimizar uma população de soluções. Dado o problema a ser otimizado, o conjunto de pesos W , número de vizinhos por indivíduo N_B e a população inicial P , o algoritmo inicializa o arquivo de soluções não-dominadas EP e o ponto ideal \mathbf{z} (Linha 1). Em G gerações, para cada solução em P , dois indivíduos pais \mathbf{x}_{n1} e \mathbf{x}_{n2} são selecionados da vizinhança (Linha 4) e uma nova solução \mathbf{y} é gerada através dos operadores genéticos *Simulated Binary Crossover* (SBX) (DEB; AGRAWAL, 1995) e mutação polinomial (DEB; GOYAL, 1996) (Linha 5). Ao avaliar os objetivos de \mathbf{y} , o ponto ideal \mathbf{z} é atualizado com os menores valores para cada objetivo encontrados até o momento (Linhas 6 e 7). Então a vizinhança é atualizada substituindo por \mathbf{y} os indivíduos que possuem valor de função de agregação maior que o novo indivíduo, considerando o seu respectivo vetor de

Algoritmo 1 MOEA/D

Entrada: MOP f a ser otimizado e o número máximo de gerações G ;
 Conjunto W de N vetores de pesos e indivíduos da população inicial P ;
 Tamanho da vizinhança N_B .

Saída: Aproximação da \mathcal{PF} para o MOP f

- 1: $EP \leftarrow \emptyset, z \leftarrow \{\infty\}^{N_O}, g \leftarrow 0$
- 2: **enquanto** $g < G$ **faça**
- 3: **para** $\mathbf{x}_i \in P$ **faça**
- 4: $\mathbf{x}_{n1}, \mathbf{x}_{n2} \leftarrow \text{SelecaoPorTorneio}(B(\mathbf{x}_i))$
- 5: $\mathbf{y} \leftarrow \text{OperadoresGeneticos}(\mathbf{x}_{n1}, \mathbf{x}_{n2})$
- 6: Avalia a nova solução \mathbf{y} para obter os objetivos $\mathbf{f}(\mathbf{y})$
- 7: $\mathbf{z} \leftarrow \{\min(z_j, f_j(\mathbf{y})) : j = 1, \dots, N_O\}$
- 8: **para** $\mathbf{x}_b \in B(\mathbf{x}_i)$ **faça**
- 9: $\mathbf{x}_b \leftarrow \mathbf{y}$, se $F(\mathbf{y}|\mathbf{w}_b, \mathbf{z}) < F(\mathbf{x}_b|\mathbf{w}_b, \mathbf{z})$
- 10: **fim para**
- 11: $EP \leftarrow \text{AtualizeArquivo}(EP, \mathbf{y})$
- 12: $g \leftarrow g + 1$
- 13: **fim para**
- 14: **fim enquanto**
- 15: **retorne** População com as soluções não dominadas de EP .

pesos (Linha 9). O arquivo de soluções não-dominadas (EP) é atualizado removendo as soluções dominadas, adicionando \mathbf{y} se esta é não dominada (Linha 11). Finalmente, o processo é repetido para cada indivíduo e, ao final das G gerações, a população das soluções não-dominadas do arquivo EP é retornada (ZHANG; LI, 2007).

Diversas funções de agregação são encontradas na literatura, podemos citar:

- soma ponderada:

$$F_{WS}(\mathbf{x}|\mathbf{w}) = \sum_{j=1}^{N_O} w_j f_j(\mathbf{x}) \quad (4)$$

- Tchebycheff (TCHE):

$$F_{TCHE}(\mathbf{x}|\mathbf{w}, \mathbf{z}) = \max_{1 \leq j \leq N_O} \left\{ \frac{f_j(\mathbf{x}) - z_j}{w_j} \right\} \quad (5)$$

- Tchebycheff normalizado (NTCHE):

$$F_{NTCHE}(\mathbf{x}|\mathbf{w}, \mathbf{z}, \bar{\mathbf{z}}) = \max_{1 \leq j \leq N_O} \left\{ \frac{f_j(\mathbf{x}) - z_j}{\bar{z}_j - z_j} * \frac{1}{w_j} \right\} \quad (6)$$

- *Penalty Boundary Intersection (PBI)*:

$$\begin{aligned}
 F_{PBI}(\mathbf{x}|\mathbf{w}, \mathbf{z}) &= d_1 + \theta d_2 \\
 d_1 &= \frac{\|(\mathbf{f}(\mathbf{x}) - \mathbf{z})^T \mathbf{w}\|}{\|\mathbf{w}\|} \\
 d_2 &= \|\mathbf{f}(\mathbf{x}) - (\mathbf{z} + d_1 \mathbf{w})\|
 \end{aligned} \tag{7}$$

Em todas as funções de agregação citadas, em problemas de minimização, valores menores indicam melhor qualidade da solução. A abordagem da soma ponderada não é capaz de encontrar todos os pontos da \mathcal{PF} se esta for não-convexa. No entanto, na abordagem TCHE sob certas condições suaves, cada ponto da \mathcal{PF} minimiza a Equação 5 para certo peso \mathbf{w} (MIETTINEN, 1999). Estudos mais recentes (DEB; JAIN, 2014; QI *et al.*, 2014) mostram empiricamente que a formulação da função de agregação TCHE na Equação 5 gera populações mais diversificadas que a formulação original $F_{TCHE}(\mathbf{x}|\mathbf{w}, \mathbf{z}) = \max_{1 \leq j \leq N_o} \{(f_j(\mathbf{x}) - z_j)w_j\}$ utilizada por Zhang e Li (2007). NTCHE normaliza os valores de cada objetivo em relação ao ponto ideal \mathbf{z} e o ponto nadir $\bar{\mathbf{z}}$, ou seja, o ponto com os maiores valores de cada objetivo. Finalmente, PBI consiste em somar as distâncias do ponto $\mathbf{f}(\mathbf{x})$ para o ponto ideal \mathbf{z} (d_1) e de $\mathbf{f}(\mathbf{x})$ para sua projeção na direção \mathbf{w} (d_2) multiplicada por um parâmetro de penalidade $\theta > 0$. Desse modo a função de agregação PBI implicitamente mantém a convergência e diversidade da população, sendo que as outras funções diversificam as soluções de acordo com a distribuição dos pesos.

2.3.1 MOEA/D-DE E EVOLUÇÃO DIFERENCIAL

Algumas variantes foram propostas para o algoritmo MOEA/D original descrito. Li e Zhang (2009) propõem o MOEA/D-DE, um algoritmo MOEA/D com algumas modificações na seleção dos indivíduos e na atualização da vizinhança. Outra diferença é que o MOEA/D-DE emprega os operadores do algoritmo Evolução Diferencial (DE, do inglês *Differential Evolution*) (STORN; PRICE, 1997) na fase de reprodução. Além disso, o MOEA/D-DE originalmente não mantém um arquivo de soluções não-dominadas.

O Algoritmo 2 descreve os passos do MOEA/D-DE. Na seleção dos indivíduos para reprodução, os indivíduos são escolhidos aleatoriamente da vizinhança, com probabilidade δ , ou da população inteira, com probabilidade $1 - \delta$ (Linha 4). Os operadores DE são utilizados na geração do novo indivíduo a partir da solução da iteração atual \mathbf{x}_i e dos vizinhos selecionados $\mathbf{x}_{n1}, \mathbf{x}_{n2}, \dots$ (Linha 6). O indivíduo gerado é comparado com seus vizinhos em ordem aleatória, substituindo até N_R indivíduos, nos casos em que o novo indivíduo é melhor, de acordo com a função de agregação e pesos utilizado (Linhas 10 e 11).

Algoritmo 2 MOEA/D-DE

Entrada: MOP \mathbf{f} a ser otimizado e o número máximo de gerações G ;
 Conjunto W de N vetores de pesos e indivíduos da população inicial P ;
 Tamanho da vizinhança N_B ;
 Probabilidade δ de escolher a vizinhança e número de substituições N_R .

Saída: Aproximação da \mathcal{PF} para o MOP \mathbf{f}

```

1:  $\mathbf{z} \leftarrow \{\infty\}^{N_O}, g \leftarrow 0$ 
2: enquanto  $g < G$  faça
3:   para  $\mathbf{x}_i \in P$  faça
4:      $B_i \leftarrow B(\mathbf{x}_i)$ , se  $\text{aleatorio}(0,1) < \delta$ , senão  $P$ 
5:      $\mathbf{x}_{n1}, \mathbf{x}_{n2}, \dots \leftarrow \text{SelecaoAleatoria}(B_i)$ 
6:      $\mathbf{y} \leftarrow \text{OperadoresEvolucaoDiferencial}(\mathbf{x}_i, \mathbf{x}_{n1}, \mathbf{x}_{n2}, \dots)$ 
7:     Avalia a nova solução  $\mathbf{y}$  para obter os objetivos  $\mathbf{f}(\mathbf{y})$ 
8:      $\mathbf{z} \leftarrow \{\min(z_j, f_j(\mathbf{y})) : j = 1, \dots, N_O\}$ 
9:     enquanto número de atualizações  $< N_R$  faça
10:      Seleciona aleatoriamente sem reposição  $\mathbf{x}_b \in B_i$ 
11:       $\mathbf{x}_b \leftarrow \mathbf{y}$ , se  $F(\mathbf{y}|\mathbf{w}_b, \mathbf{z}) < F(\mathbf{x}_b|\mathbf{w}_b, \mathbf{z})$ 
12:    fim enquanto
13:  fim para
14:   $g \leftarrow g + 1$ 
15: fim enquanto
16: retorne  $P$ .
```

DE é um algoritmo de busca estocástica baseado em populações proposto por Storn e Price (1997). Os operadores DE possuem dois parâmetros: a taxa de cruzamento (CR) e um fator de escalarização (F). Uma nova solução é gerada através da diferença entre pares de indivíduos selecionados ($\mathbf{x}_{n1} - \mathbf{x}_{n2}, \mathbf{x}_{n3} - \mathbf{x}_{n4}, \dots$), multiplicada por F e combinada com um indivíduo alvo (\mathbf{x}_p). Na proposta do algoritmo MOEA/D-DE (LI; ZHANG, 2009), novas soluções são geradas com o operador $DE/rand/1/bin$ (descrito no próximo parágrafo) e o vetor alvo é sempre o indivíduo da iteração atual ($\mathbf{x}_p = \mathbf{x}_i$). Após a geração do novo indivíduo pelos operadores DE, a mutação polinomial é realizada em cada variável com probabilidade $1/N_D$ (DEB; GOYAL, 1996).

As variantes dos operadores de reprodução do algoritmo DE são denotadas por $DE/a/b/c$, onde a é o modo como os pais são selecionados, b é o número de pares utilizados e c é o modo como o cruzamento com o vetor alvo Vx_t é realizado. Assim, o operador $DE/rand/1/bin$ seleciona aleatoriamente ($a = rand$) um ($b = 1$) par de indivíduos, \mathbf{x}_{n1} e \mathbf{x}_{n2} , e a nova solução ($\mathbf{y} = \{y_0, \dots, y_{N_D}\}$) é gerada com o cruzamento binomial ($c = bin$), i.e., a probabilidade de escolher um componente de \mathbf{x}_t segue uma distribuição binomial:

$$y_d = \begin{cases} x_{p,d} + (x_{n1,d} - x_{n2,d}) * F & \text{se } \text{aleatorio}(0,1) < CR \text{ ou } r = i \\ x_d^p & \text{caso contrário} \end{cases} \quad (8)$$

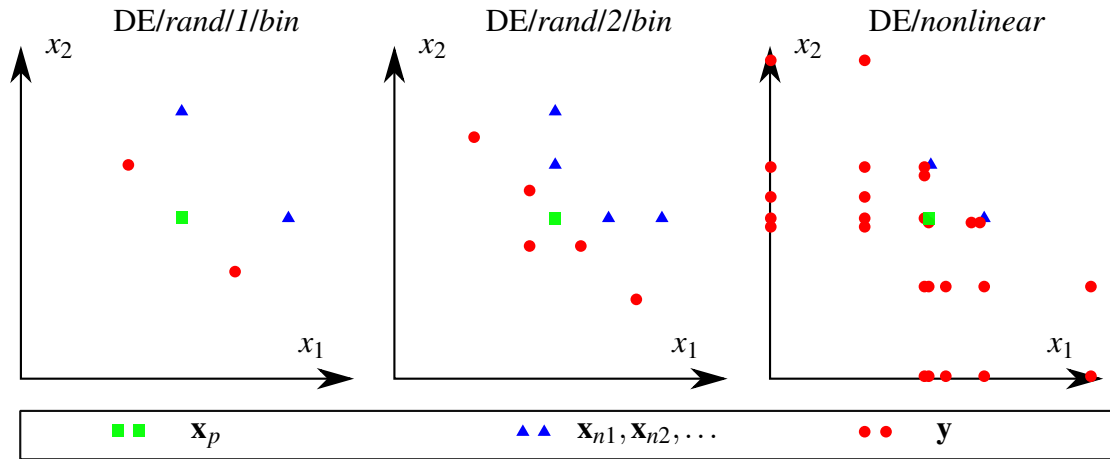


Figura 3: Exemplo de soluções geradas pelos operadores *DE/rand/1/bin*, *DE/rand/2/bin* e *DE/nonlinear* com $CR = 1.0$ e $F = 0.5$.

onde r é um valor aleatório inteiro entre 1 e N_D . Naturalmente, no operador *DE/rand/2/bin*, a nova solução ($\mathbf{y} = \{y_0, \dots, y_{N_D}\}$) é gerada através da fórmula:

$$y_d = \begin{cases} x_{p,d} + (x^{n1,d} - x^{n2,d} + x^{n3,d} - x^{n4,d}) * F & \text{se } \text{aleatorio}(0, 1) < CR \text{ ou } r = i \\ x_{p,d} & \text{caso contrário} \end{cases} \quad (9)$$

Finalmente, o operador aqui denominado *DE/nonlinear* foi proposto por Sindhya *et al.* (2011) e será testado por possuir melhor desempenho em problemas com curvas e interdependências não-lineares das soluções no espaço de decisão. A nova solução é gerada com o polinômio:

$$\mathbf{y} = \frac{(\mathbf{x}_p - 2\mathbf{x}_{n1} + \mathbf{x}_{n2})}{2} \mathbf{v}^2 + \frac{(4\mathbf{x}_{n1} - 3\mathbf{x}_p - \mathbf{x}_{n2})}{2} \mathbf{v} + \mathbf{x}_p \quad (10)$$

onde \mathbf{v} é um vetor de N_D variáveis aleatórias uniformes, entre $[0; 2]$ com probabilidade 0,75 ou entre $[2; 3]$ com probabilidade 0,25. Os três operadores possuem diferentes características na exploração do espaço. A Figura 3 apresenta exemplos de soluções geradas com os três operadores em dada distribuição de soluções pais ($\mathbf{x}_p, \mathbf{x}_{n1}, \mathbf{x}_{n2}, \dots$).

2.3.2 MOEA/D-STM

O algoritmo MOEA/D original, bem como sua variante MOEA/D-DE, tratam cada sub-problema igualmente e cada um recebendo o mesmo esforço computacional, gerando indivíduos e atualizando sua vizinhança por um mesmo número de gerações. Além disso, considerando os subproblemas como agentes que minimizam a função de agregação e as soluções como agentes que minimizam a distância para seus vetores de pesos relacionados, existe um conflito de interesse que é ignorado nos algoritmos MOEA/D e MOEA/D-DE.

O *MOEA/D with Stable Matching* (MOEA/D-STM) é uma variante do algoritmo MOEA/D que procura evitar os problemas de desbalanceamento de esforço computacional e conflito de interesse entre subproblemas e soluções. Para isso, o MOEA/D-STM utiliza a estratégia de alocação dinâmica de recursos, originalmente proposta para a variante *MOEA/D with Dynamic Resource Allocation* (MOEA/D-DRA) (ZHANG *et al.*, 2009), para valorizar a seleção de subproblemas mais difíceis (com maior demanda iterações do algoritmo) de acordo com seus valores de utilidade. Quanto ao conflito entre soluções e subproblemas, durante a atualização da população, o MOEA/D-STM faz uso do *Stable Matching* ou pareamento estável. *Stable Matching* é uma estratégia proposta nas ciências econômicas, onde cada solução é associada a um subproblema considerando ambos os interesses de modo que as associações sejam estáveis (LI *et al.*, 2014b). O procedimento do MOEA/D-STM é descrito no Algoritmo 4 e a seleção baseada no pareamento estável é mostrada no Algoritmo 3.

No começo de cada geração, a população selecionada para a evolução (I) é formada por um quinto da população original ($\lfloor N/5 \rfloor$): N_O indivíduos cujo subproblema é otimizar individualmente algum objetivo do MOP (ou cujos pesos são colineares ao eixo de algum objetivo) e $\lfloor N/5 \rfloor - N_O$ indivíduos selecionados por um torneio de 10 rodadas sem reposição, baseando-se nos valores de utilidade. Os valores de utilidade (π^i) de cada indivíduo são inicializados com 1 e a cada 30 gerações são atualizados de acordo com a equação:

$$\pi^i = \begin{cases} 1 & \text{se } \Delta(\mathbf{x}_i^g, \mathbf{x}_i^{g-30}) > 0.001 \\ (0.95 + 0.05 \frac{\Delta(\mathbf{x}_i^g, \mathbf{x}_i^{g-30})}{0.001}) \pi^i & \text{caso contrário} \end{cases} \quad (11)$$

onde \mathbf{x}_i^{g-30} é a solução \mathbf{x}_i^g na geração $(g - 30)$ e $\Delta(\mathbf{x}_a, \mathbf{x}_b)$ é o decréscimo relativo da função de agregação para o indivíduo \mathbf{x}_a , definido para problemas de minimização como:

$$\Delta(\mathbf{x}_a, \mathbf{x}_b) := \frac{F(\mathbf{x}_b | \mathbf{w}_a, \mathbf{z}) - F(\mathbf{x}_a | \mathbf{w}_a, \mathbf{z})}{F(\mathbf{x}_b | \mathbf{w}_a, \mathbf{z})} \quad (12)$$

O pareamento subproblemas-soluções é dito estável se: (1) para uma solução que não possui um par com subproblema, nenhum subproblema prefere aquela solução do que sua solução atual; (2) uma solução \mathbf{x} que está associada a um subproblema diferente de \mathbf{w} , \mathbf{x} prefere seu problema atual ao invés de \mathbf{w} e \mathbf{w} prefere sua solução atual ao invés de \mathbf{x} , ou seja, a troca de pares associados nunca será vantajosa para ambos. O Algoritmo 3 realiza o pareamento entre subproblemas e soluções. Enquanto há subproblemas sem solução par, um deles é selecionado aleatoriamente. Busca-se a solução preferível para o subproblema (de acordo com as Equações 5 a 7), que ele ainda não tentou formar par. Se a solução encontrada está livre (não possui par com nenhum subproblema), ela é escolhida e forma-se o par. Caso contrário, forma-se o par

com o melhor dentre os dois subproblemas (aquele considerado e o subproblema atualmente associado com a solução escolhida), de acordo com a preferência da solução (Equação 13). Cada subproblema possui uma lista tabu das soluções com que já tentou formar par em tentativas anteriores. A preferência da solução ($pref_x$) é baseada na distância projetada dos seus objetivos normalizados ($\bar{\mathbf{f}}(\mathbf{x}_j)$) até o vetor peso do problema referente:

$$pref_x(\mathbf{x}, \mathbf{w}) = \bar{\mathbf{f}}(\mathbf{x}) - \frac{\mathbf{w}^T \bar{\mathbf{f}}(\mathbf{x})}{\|\mathbf{w}\|} \mathbf{w} \quad (13)$$

e a preferência dos subproblemas é dada pelo valor da função de agregação utilizada (Equações 4-7). Pode ser demonstrado que o Algoritmo 3 converge, o pareamento é estável e empiricamente gera populações com mais diversidade (LI *et al.*, 2014b).

Algoritmo 3 SeleçãoSTM

Entrada: População $P = \{(\mathbf{x}_1, \mathbf{f}(\mathbf{x}_1)), \dots, (\mathbf{x}_{1.2N}, \mathbf{f}(\mathbf{x}_{1.2N}))\}$;

Pesos para os subproblemas $\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_N\}$.

Saída: Soluções de P que formam pareamento estável com os subproblemas de \mathbf{W}

- 1: $P^* \leftarrow \emptyset$
 - 2: **enquanto** ainda existem subproblemas sem par **faça**
 - 3: $\mathbf{w}_i \leftarrow$ seleciona aleatoriamente um subproblema sem par
 - 4: $\mathbf{x}_j \leftarrow$ seleciona a solução mais preferível ($\min_{\mathbf{x}} \{F(\mathbf{x}|\mathbf{w}, \mathbf{z}) : \mathbf{x} \in P\}$), que \mathbf{w}_i ainda não tentou formar par
 - 5: **se** \mathbf{x}_j não possui par **então**
 - 6: $P^* \leftarrow P^* \cup \{(\mathbf{w}_i, \mathbf{x}_j)\}$
 - 7: **senão**
 - 8: **se** $pref_x(\mathbf{x}_j, \mathbf{w}_i) < pref_x(\mathbf{x}_j, \mathbf{w}_k)$ **então**
 - 9: $P^* \leftarrow (P^* \setminus \{(\mathbf{w}_k, \mathbf{x}_j)\}) \cup \{(\mathbf{w}_i, \mathbf{x}_j)\}$ // \mathbf{x}_j prefere \mathbf{w}_i ao invés de seu par atual \mathbf{w}_k
 - 10: **fim se**
 - 11: **fim se**
 - 12: **fim enquanto**
 - 13: **retorne** População com soluções \mathbf{x} e objetivos $\mathbf{f}(\mathbf{x})$ de P^* .
-

Os algoritmos MOEA/D, MOEA/D-DE e MOEA/D-STM descritos serão testados neste trabalho para compor a plataforma ELMOEA/D para otimização de MOPs com funções-objetivo custosas. Em resumo:

- O MOEA/D decompõe o MOP em subproblemas definidos por vetores de pesos \mathbf{W} e mantém soluções compondo uma população P . Com informações de soluções para subproblemas semelhantes, ou vizinhança, novas soluções são geradas com operadores genéticos e a vizinhança é atualizada conforme os valores da função de agregação;
- O MOEA/D-DE gera novas soluções com operadores do algoritmo DE e pode utilizar informações dos indivíduos de toda a população com probabilidade $1 - \delta$. A atualização

Algoritmo 4 MOEA/D-STM

Entrada: MOP \mathbf{f} a ser otimizado e o número máximo de gerações G ;
 Conjunto de N pesos W e indivíduos da população inicial P ;
 Tamanho da vizinhança N_B ;
 Probabilidade de escolher a vizinhança δ e número de substituições N_R .

Saída: Aproximação da \mathcal{PF} para o MOP \mathbf{f}

```

1:  $\mathbf{z} \leftarrow \{\infty\}^{N_O}$ ,  $\boldsymbol{\pi} \leftarrow \mathbf{1}$ ,  $g \leftarrow 0$ 
2: enquanto  $g < G$  faça
3:    $I \leftarrow \{\mathbf{x}'_1, \dots, \mathbf{x}'_{N_O} : \mathbf{x}'_j \text{ otimiza o objetivo } j\} \cup \{\mathbf{x}'_1, \dots, \mathbf{x}'_{\lfloor N/5 \rfloor - N_O} : \mathbf{x}' \in \text{Torneio10}(P, \boldsymbol{\pi})\}$ 
4:    $Q \leftarrow \emptyset$ 
5:   para  $\mathbf{x}_i \in I$  faça
6:      $B_i \leftarrow B(\mathbf{x}_i)$ , se  $\text{aleatorio}(0, 1) < \delta$ , senão  $P$ 
7:      $\mathbf{x}_{n1}, \mathbf{x}_{n2}, \dots \leftarrow \text{SelecaoAleatoria}(B_i)$ 
8:      $\mathbf{y} \leftarrow \text{OperadoresEvolucaoDiferencial}(\mathbf{x}_i, \mathbf{x}_{n1}, \mathbf{x}_{n2}, \dots)$ 
9:     Avalia a nova solução  $\mathbf{y}$  para obter os objetivos  $\mathbf{f}(\mathbf{y})$ 
10:     $\mathbf{z} \leftarrow \{\min(z_j, f_j(\mathbf{y})) : j = 1, \dots, N_O\}$ 
11:     $Q \leftarrow Q \cup \{\mathbf{y}, \mathbf{f}(\mathbf{y})\}$ 
12:  fim para
13:   $P \leftarrow \text{STM}(P \cup Q)$ 
14:   $g \leftarrow g + 1$ 
15:   $\boldsymbol{\pi} \leftarrow \text{AtualizaUtilidades}(P)$ , se  $g$  é múltiplo de 30
16: fim enquanto
17: retorne  $P$ .
```

da vizinhança (ou de toda a população) é feita substituindo até N_R indivíduos com valor de função de agregação inferior à nova solução;

- Finalmente, o MOEA/D-STM considera que subproblemas mais difíceis devem ser favorecidos na reprodução e que existe conflito entre soluções e subproblemas durante a atualização da população. Para tal, os algoritmos de alocação dinâmica de recursos e pareamento estável são utilizados. Ressaltando que isso torna o algoritmo mais complexo e com custo computacional significativamente maior (pior caso de $O(NN_O \log(N_O))$ por seleção) (LI *et al.*, 2014b).

2.3.3 MOEA/D-RBF

O algoritmo *MOEA/D assisted by RBF surrogates* (MOEA/D-RBF), proposto por Martínez e Coello (2013), integra o MOEA/D original (Algoritmo 1) com redes neurais tipo *Radial Basis Function* (RBFs) em uma plataforma para otimização evolutiva de MOPs custosos. Ainda, MOEA/D-RBF mantém dois arquivos de soluções, as soluções avaliadas com a função custosa e um arquivo de treinamento. O procedimento executado pelo MOEA/D-RBF consiste em:

Inicialização: o arquivo de treinamento é inicializado com soluções aleatórias que também são adicionadas no arquivo de soluções avaliadas. Até que o critério de parada seja satisfeito, o algoritmo itera nos seguintes passos;

Construção do *surrogate*: três redes RBF com diferentes funções de ativação (sigmoidal, gaussiana e *multiquadric*) em um *ensemble* cooperativo são treinadas nesta fase. Os pesos de entrada são determinados pelo algoritmo de clusterização *k-means* e os pesos de saída são ajustados por um algoritmo DE que minimiza o erro dos modelos em prever a saída de cada função. As três redes cooperam de modo que a saída é a soma ponderada das saídas de cada rede, com pesos somando 1 e inversamente proporcionais ao erro de treinamento de cada modelo;

Busca por soluções: o algoritmo MOEA/D busca soluções promissoras que otimizem as aproximações das redes RBF;

Seleção de pontos para avaliação: soluções promissoras bem distribuídas são selecionadas da população final encontrada pelo MOEA/D. Um conjunto menor de vetores de pesos separa os vetores de pesos do MOEA/D em subconjuntos distintos por um critério de distância. A melhor solução de cada subconjunto, desde que diferente de qualquer outra solução já avaliada, é selecionada para avaliação com a função-objetivo original;

Atualização dos arquivos: as soluções avaliadas são adicionadas aos arquivos de soluções avaliadas e de treinamento. Como o arquivo de treinamento possui tamanho limitado, cada solução é associada a um vetor de pesos e a nova solução \mathbf{x}^* substitui aquela associada ao peso na qual \mathbf{x}^* possui o melhor valor para a função de agregação. Caso a nova solução não seja melhor que a solução presente no arquivo, a solução mais próxima é substituída;

Atualização da população: a população inicial do MOEA/D é reiniciada a partir de informações sobre as melhores soluções avaliadas.

Como a plataforma do MOEA/D-RBF inspirou a proposta do ELMOEAD, esta será usada como abordagem de comparação e os passos em comum serão descritos com mais detalhes na Seção 3, que também salienta as principais diferenças entre os dois algoritmos.

2.4 SELEÇÃO ADAPTATIVA DE OPERADORES

Cada heurística ou operador (cruzamento e/ou mutação) sozinho valoriza diferentes aspectos da busca. A escolha da heurística certa pode demandar tempo, principalmente quando

estes aspectos são pouco explorados ou muito atrelados ao problema. A seleção adaptativa de operadores refere-se ao uso de um método ou mecanismo de aprendizado para seleção de operadores na resolução de problemas de busca. Com isso, o espaço de busca da seleção adaptativa compreende todas as heurísticas possíveis de serem selecionadas (*pool* de operadores) e o processo de busca pelo melhor operador pode ser feito antes (*offline*) ou durante (*online*) a busca por soluções (BURKE *et al.*, 2013).

Probability Matching (PM) é um mecanismo de seleção por roleta que pode ser empregado na seleção adaptativa de operadores *online* (GOLDBERG, 1990). No começo da busca, todos os N_{Op} operadores disponíveis têm a mesma probabilidade de serem selecionados ($p_k^0 = 1/N_{Op}, \forall k \in \{1, \dots, N_{Op}\}$), esta probabilidade p_k é atualizada para cada operador com:

$$\begin{aligned} q_k^{g+1} &= (1 - \alpha)q_k^g + \alpha r_k \\ p_k^{g+1} &= p_{min} + (1 - N_{Op} + p_{min}) \frac{q_k^{g+1}}{\sum_{l=1}^{N_{Op}} q_l^{g+1}} \end{aligned} \quad (14)$$

onde q_k^g é a qualidade da solução na geração g , r_k é a recompensa quanto ao desempenho do operador k , p_{min} e α são parâmetros dados, referentes à probabilidade mínima de cada operador ser selecionado e a taxa de aprendizado, respectivamente.

Neste trabalho, a recompensa é calculada baseando-se no decaimento relativo do *fitness* durante a atualização da população, i.e., $r_k = \Delta(\mathbf{y}, \mathbf{x}_i)$ onde \mathbf{y} é a solução candidata, \mathbf{x}_i é a solução pai da iteração atual e $\Delta(\cdot, \cdot)$ é o decréscimo relativo da função de agregação definido na Equação 12.

Multi-armed bandit (AUER *et al.*, 2002) também trata da seleção de operadores *online*, porém é determinística e considera que a melhor estratégia de seleção é aquela que maximiza a recompensa acumulada pelos operadores ao longo do tempo. Em particular, a *Fitness-Rate-Rank-Based Multi-armed Bandit* (FRRMAB) considera o comportamento dinâmico da busca e usa uma estrutura de janela deslizante de tamanho fixo N_{sw} para guardar as recompensas de cada operador (LI *et al.*, 2014a). Assim, o operador selecionado será aquele que maximiza a expressão:

$$FRR_k^t + C_s \left(\frac{2 \ln \sum_{l=1}^{N_{Op}} n_l^t}{n_k^t} \right) \quad (15)$$

sendo que C_s é o parâmetro de escalarização que controla a exploração do espaço de possibilidades, n_k^t é o número de vezes que o operador k aparece nos dados da janela deslizante e o valor de

crédito para o operador k (FRR_k^t) é calculado com:

$$Decay_k = D^{rank_k} r_k$$

$$FRR_k^t = \frac{Decay_k}{\sum_{l=1}^{N_{Op}} Decay_l} \quad (16)$$

onde D é um parâmetro de decaimento que controla a influência do melhor operador e $rank_k$ é a ordem do operador k , quanto a soma das recompensas de cada operador.

As estratégias PM e FRRMAB são utilizadas na escolha dos operadores DE citados na Seção 2.3.1 (DE/rand/1/bin, DE/rand/2/bin, DE/nonlinear), em conjunto com o algoritmo MOEA/D-DE. Para mais detalhes sobre o uso do MOEA/D-DE com seleção adaptativa de operadores, ver (LI *et al.*, 2014a).

2.5 EXTREME LEARNING MACHINE (ELM)

Redes neurais artificiais são formadas por conjuntos de unidades de processamento (neurônios) conectadas por pesos e podem ser utilizadas para classificação e regressão. A tarefa de regressão consiste em aprender funções contínuas, de modo que o modelo possa relacionar um valor objetivo (saídas) ao estado que lhe deu origem (entradas). Em redes neurais, este processo é realizado adaptando seus parâmetros (pesos), sendo que o treinamento chamado “supervisionado” necessita das saídas e entradas associadas (conjunto de treinamento) (RUSSELL; NORVIG, 2003).

As máquinas de aprendizado extremo, ou *Extreme Learning Machines* (ELMs) foram propostas por Huang *et al.* (2004) e recentemente generalizadas como *Single Layer Feedforward Networks* (HUANG *et al.*, 2012). Assim como ilustrado na Figura 4, quando aplicada uma entrada \mathbf{x} , o valor de saída para o j -ésimo neurônio de saída da rede ELM é dado por:

$$f_{ELM,j}(\mathbf{x}) = \sum_{k=1}^{N_H} h_k(\mathbf{x}) \beta_{k,j} = \mathbf{h}(\mathbf{x}) \boldsymbol{\beta}_j \quad (17)$$

onde $h(\cdot)$ é a função de ativação para os N_H neurônios da camada oculta e $\boldsymbol{\beta}_j$ são os pesos para o neurônio j na saída da rede. A função de ativação $h(\mathbf{x})$ pode ser qualquer função contínua por partes $h_k(\mathbf{x}) = h(\mathbf{x}, \mathbf{a}_k, b_k)$, que toma como argumentos a entrada \mathbf{x} , os pesos de entrada \mathbf{a}_k e o *bias* b_k . As abordagens mais comuns são:

- Sigmoidal (SIG):

$$h_{SIG}(\mathbf{x}, \mathbf{a}, b) = \frac{1}{1 + \exp(-(\mathbf{a} \cdot \mathbf{x} + b))} \quad (18)$$

- Gaussiana (GAU):

$$h_{GAUSS}(\mathbf{x}, \mathbf{a}, b) = \exp(-b \|\mathbf{x} - \mathbf{a}\|) \quad (19)$$

- *Multiquadric* (MQ):

$$h_{MULT}(\mathbf{x}, \mathbf{a}, b) = (\|\mathbf{x} - \mathbf{a}\|^2 + b^2)^{1/2} \quad (20)$$

Cada função de ativação possui características e sua escolha depende do conjunto de dados de treinamento. A Figura 5 mostra o formato das três funções de ativação (SIG, GAU e MQ), conforme o valor da entrada \mathbf{x} .

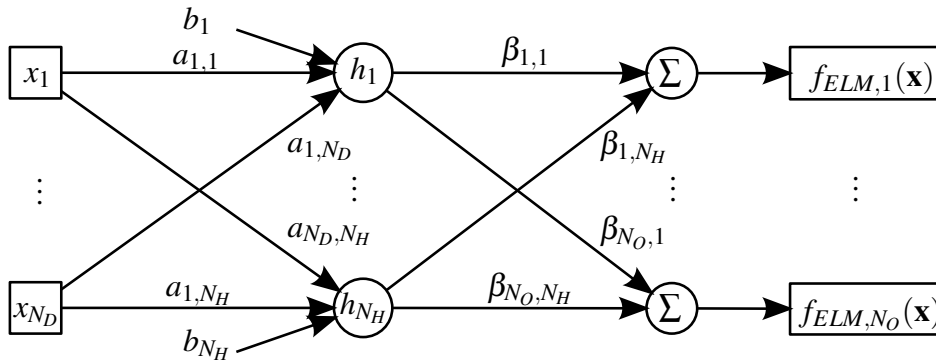


Figura 4: ELM como uma rede neural de camada intermediária única e múltiplas saídas: a função de uma saída $f_{ELM,1}(\mathbf{x})$ para a entrada $\mathbf{x} = (x_1, \dots, x_{N_D})$ será o produto entre a saída dos neurônios ocultos (h_1, \dots, h_{N_H}) e os pesos de saída $(\beta_{1,1}, \dots, \beta_{N_H,1})$, dados os pesos de entrada $(a_{11}, \dots, a_{N_D N_H})$ e o bias de cada neurônio (b_1, \dots, b_{N_H}) .

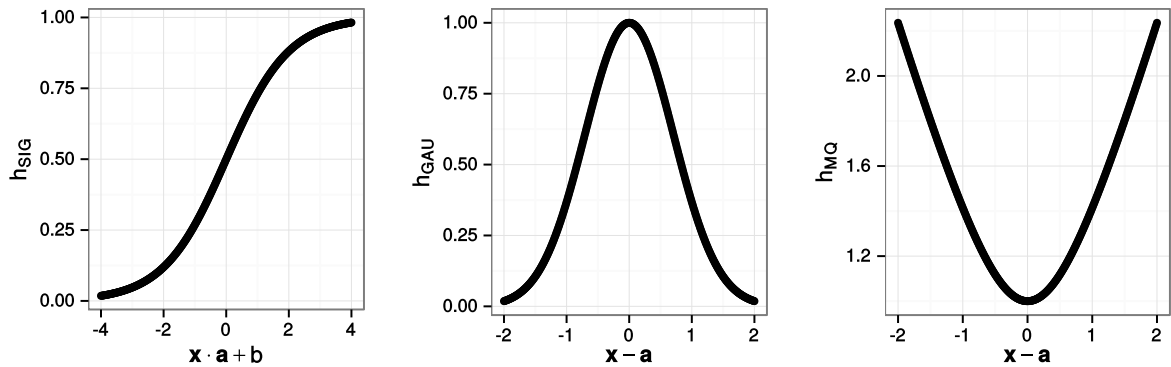


Figura 5: Formato de cada função de ativação (SIG, GAU, MQ) para dadas entradas \mathbf{x} . Nas funções GAU e MQ, o valor de bias $b = 1$.

Segundo Bartlett (1998), quanto menor é a norma dos pesos de saída da rede, melhor é sua capacidade de generalização. As ELMs consideram esta teoria, minimizando o erro de treinamento e a norma dos pesos de saída. Diferente de outros métodos de aprendizado, ELMs atribuem valores aleatórios para os parâmetros $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_K$ e b_1, b_2, \dots, b_{N_H} e ajustam apenas os pesos de saída. Dadas N_T entradas do conjunto de treinamento, $\chi = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_T})$, e

suas respectivas saídas, $\mathbf{v}^T = (y_1, y_2, \dots, y_{N_T})$, os pesos da ELM são ajustados de acordo com (HUANG *et al.*, 2004):

$$\boldsymbol{\beta} = \mathbf{H}^+ \mathbf{v} \quad (21)$$

onde \mathbf{H} é a matriz com todas as saídas dos neurônios ocultos para cada entrada:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_{N_T}) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & \dots & h_{N_H}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_{N_T}) & \dots & h_{N_H}(\mathbf{x}_{N_T}) \end{bmatrix},$$

e \mathbf{H}^+ é a pseudoinversa de Moore-Penrose. Huang *et al.* (2012) propõem uma versão regularizada na qual a otimização dos pesos de saída é feita através do problema com restrições:

$$\begin{aligned} \text{Minimizar}_{\{\boldsymbol{\beta}, \boldsymbol{\xi}\}} \quad Z &= \frac{1}{2} \|\boldsymbol{\beta}\|^2 + C \frac{1}{2} \sum_{i=1}^{N_T} \xi_i^2 \\ \text{sujeito a} \quad \mathbf{h}(\mathbf{x}_i) \boldsymbol{\beta} &= y_i - \xi_i \quad i = 1, \dots, N_T. \end{aligned}$$

em que C é um parâmetro regularizador dado. Aplicando o teorema Karush–Kuhn–Tucker e derivando o problema dual, obtemos para o caso $N_T > N_H$ (HUANG *et al.*, 2012):

$$\boldsymbol{\beta} = \left(\mathbf{H}^T \mathbf{H} + \frac{\mathbf{I}}{C} \right)^{-1} \mathbf{H}^T \mathbf{v}. \quad (22)$$

e caso $N_T < N_H$, é desejável resolver a inversa de tamanho $N_T \times N_T$ e calcular os pesos de saída com:

$$\boldsymbol{\beta} = \mathbf{H}^T \left(\mathbf{H} \mathbf{H}^T + \frac{\mathbf{I}}{C} \right)^{-1} \mathbf{v}. \quad (23)$$

O treinamento da ELM ocorre em apenas um passo para resolver a Equação 22 ou Equação 23, e pode ser facilmente generalizado para uma rede com várias saídas. Assim, as ELMs tornaram-se populares devido ao baixo custo computacional. Além disso, poucos parâmetros precisam ser especificados (função de ativação, C e número de neurônios) e, dado um número suficiente de neurônios ocultos, apenas o parâmetro C precisa ser ajustado de acordo com o problema (HUANG *et al.*, 2012).

2.6 PROBLEMA DA PREDIÇÃO DE ESTRUTURAS DE PROTEÍNAS (PPEP)

Proteínas são macromoléculas que desempenham várias funções importantes em organismos vivos. Em química e biologia molecular, proteínas são polímeros lineares, constituídos de uma sequência de aminoácidos e unidos por ligações peptídicas. O termo proteínas geralmente refere-se a moléculas grandes com 50 aminoácidos ou mais. Outro termo também utilizado para

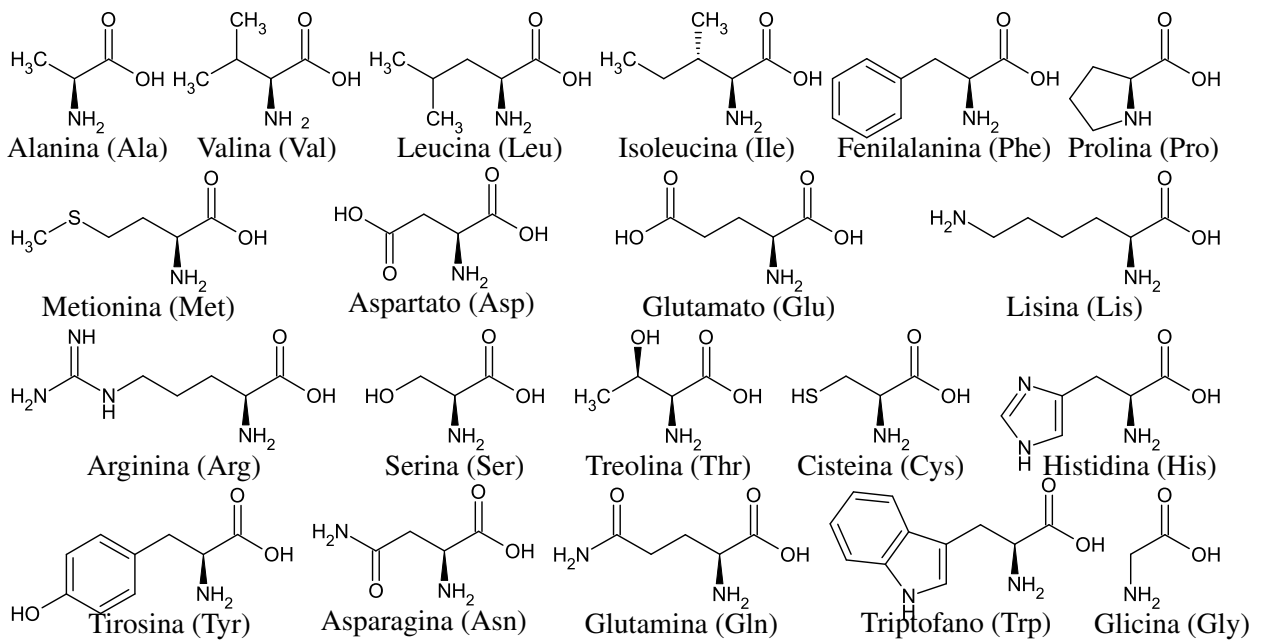


Figura 6: Estrutura dos 20 aminoácidos que ocorrem naturalmente.

moléculas pequenas (menores que 50 aminoácidos) é peptídeo. A Figura 6 mostra a estrutura molecular dos 20 aminoácidos encontrados na natureza e suas nomenclaturas. A ligação peptídica é a união do grupo amino N(H) de um aminoácido com o grupo carboxila C(=O) de outro (Figura 7). A sequência formada pelo grupo amino, carbono alfa e carboxila é chamada de *backbone* e os grupos restantes de cada aminoácido conectado são chamados de resíduos (R_j) da proteína (XU *et al.*, 2007) (Figura 7).

A Figura 7 mostra a representação da proteína nos diferentes níveis estruturais. A sequência de aminoácidos identificando a proteína é chamada de estrutura primária (Figura 7a). Devido a interações entre os aminoácidos, algumas estruturas regulares e comuns a várias proteínas podem ser observadas, como hélices- α , folha- β (Figura 7b). A estrutura terciária, o principal interesse desse trabalho, trata da conformação (dobramento) da proteína no espaço tridimensional (Figura 7c). Por fim, a estrutura quaternária da proteína consiste em dividir a estrutura completa em regiões funcionais, geralmente conjuntos compactos que dobram-se independentemente chamados domínios (Figura 7d).

O PPEP consiste em determinar a estrutura terciária, dada sua composição em aminoácidos (estrutura primária). Métodos como Difração de Raio X e Espectroscopia de Ressonância Magnética Nuclear são utilizados para descobrir experimentalmente a estrutura das proteínas, porém é um processo custoso e demorado, considerando a eficiência em que os pesquisadores descobrem novas sequências de aminoácidos (XU *et al.*, 2007). Ao comprovar que a estrutura primária contém as informações necessárias para derivar sua estrutura terciária, vários métodos

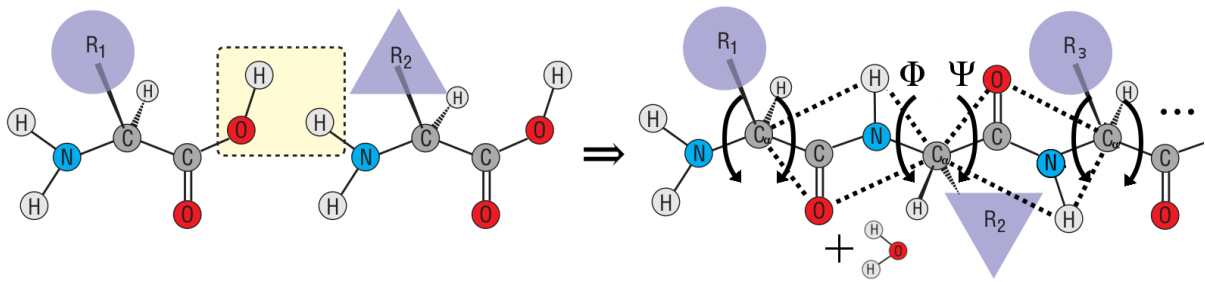


Figura 7: Ligação peptídica entre aminoácidos, planos coplanares da ligação (tracejado) e ângulos livres Φ e Ψ (adaptado de (PETSKO; RINGE, 2004)).

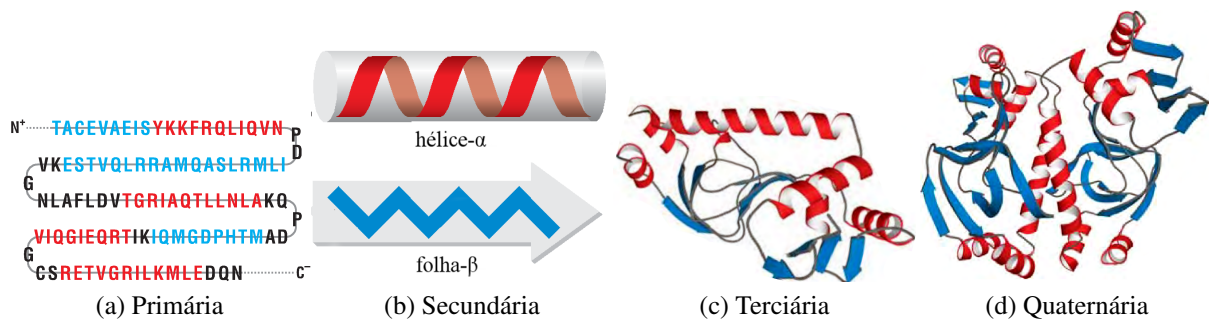


Figura 8: Representação de uma proteína em diferentes níveis estruturais (adaptado de Petsko e Ringe (2004)).

computacionais foram propostos (TSIGELNY, 2002; TRAMONTANO, 2006; XU *et al.*, 2007; BUJNICKI, 2008). Estes métodos podem ser classificados em:

Métodos baseados em padrões (homologia e *threading*): estes métodos utilizam estruturas já conhecidas e similaridades entre a proteína que se deseja descobrir a estrutura. Estes métodos obtêm sucesso na maioria dos casos, principalmente quando as estruturas conhecidas possuem relacionamento evolutivo com a proteína em questão. Porém, quando a proteína possui baixa similaridade com qualquer outra estrutura conhecida, estes métodos não garantem bom desempenho;

Métodos *ab initio*: que fazem a predição utilizando apenas informações da estrutura primária. Geralmente, os métodos *ab initio* utilizam uma função de energia global da proteína que possui valor mínimo (ou mínimo local) quando a conformação se encontra em sua estrutura final.

Este trabalho trata de um método *ab initio* para prever a estrutura de proteínas utilizando apenas sua estrutura primária, ou seja, a sequência de aminoácidos. Em particular, o método busca no espaço de valores possíveis para os ângulos livres (diedrais e ângulos livres de cada resíduo) da estrutura de forma a minimizar a energia total. A função de energia a ser

minimizada é detalhada a seguir.

2.6.1 CAMPO DE FORÇA CHARMM

Um campo de força, ou *forcefield*, especifica os parâmetros e formulações utilizadas no cálculo da energia total da molécula. Não há evidências sobre qual seria o melhor campo de força para o PPEP em todos os casos e os mais utilizados na literatura são AMBER, GROMOS e CHARMM (PONDER; CASE, 2003). Neste trabalho, utilizamos o campo de força CHARMM, em um método *ab initio* para minimizar a energia dada por:

$$\begin{aligned}
 E_C = & \sum_{ligacoes} K_b(b_C - b_{eq})^2 + \\
 & \sum_{angulos} K_\theta(\theta_C - \theta_{eq})^2 + \\
 & \sum_{diedrais} K_\Phi[1 + \cos(n\Phi_C - \gamma)] + \\
 & \sum_{Urey-Bradley} K_u(u - u_0)^2 + \\
 & \sum_{improprios} K_\xi(\xi - \xi_0)^2 + \\
 & \sum_{\phi, \psi} V_{CMAP} + \\
 & \sum_{nao-ligados} \left[\frac{A_{ij}}{r_{ij,c}^{12}} - \frac{B_{ij}}{r_{ij,c}^6} + \frac{q_i q_j}{\epsilon_0 r_{ij,c}} \right]
 \end{aligned} \tag{24}$$

As energias calculadas pelo CHARMM correspondentes aos termos ligados são:

energia de ligação: para todos os átomos que compartilham elétrons (ligações covalentes simples ou duplas), calculada pela fórmula $E_{ligacoes} = \sum K_b(b - b_{eq})^2$, em que b é o comprimento da ligação, b_{eq} é o comprimento de equilíbrio da ligação e K_b é a constante de energia, que é maior quanto mais forte a ligação;

energia dos ângulos: para cada ângulo entre as ligações, dada por $E_{angulos} = \sum K_\theta(\theta - \theta_{eq})^2$ tal que θ é o ângulo da ligação, θ_{eq} é o ângulo de equilíbrio da ligação e K_θ é a constante de energia;

energia dos ângulos diedrais: para cada ângulo diedro (entre planos), calculada por $E_{diedrais} = \sum_{n=1}^{N_{min}} K_\Phi[1 + \cos(n\Phi - \gamma)]$ tal que K_Φ é a constante para ângulos de diedrais, N_{min} é o número de mínimos, γ é o deslocamento angular;

Urey-Bradley: usado para considerar as interações entre átomos ligados a um átomo comum, como somatório de valores $E_{UB} = \sum K_u(u - u_0)^2$, onde u é a distância entre dois átomos separados por duas ligações covalentes, u_0 é a distância de equilíbrio e K_u é a constante de força de Urey-Bradley;

potencial diedral impróprio: mantém a estrutura tridimensional dos tetraedros formados quando um átomo é ligado a três outros átomos, calculando a energia gerada em casos de oscilações na estrutura com a fórmula $E_{improp} = \sum K_{\xi}(\xi - \xi_0)^2$, em que ξ é o ângulo entre os planos do tetraedro, ξ_0 é o ângulo de equilíbrio e K_{ξ} uma constante definida.

Finalmente, as energias dos átomos não-ligados, incluídas na Equação 24 são:

lei de Coulomb: para interações eletrostáticas, calculada com $E = \sum \frac{q_i q_j}{\epsilon_0 \epsilon_r r_{ij}}$, onde q_i e q_j são as cargas, r_{ij} a distância entre as cargas, ϵ_r a permissividade e ϵ_0 é a constante dielétrica do meio;

energia de dispersão: para medir a repulsão entre átomos específicos, com valor $\frac{-B_{ij}}{r_{ij}^6}$ para B_{ij} estimado com experimentos em moléculas pequenas;

energia de repulsão: mede a atração entre átomos com $\frac{A_{ij}}{r_{ij}^{12}}$, para A_{ij} estimado previamente, que junto com a energia de dispersão formam as atrações de van der Waals $E_{VW} = \sum (\frac{A_{ij}}{r_{ij,C}^{12}} - \frac{B_{ij}}{r_{ij,C}^6})$.

Além destas energias, o valor dado por V_{CMAP} na Equação 24 refere-se a uma correção dos ângulos diedrais Φ e Ψ (Figura 7).

2.6.2 FORMULAÇÃO DO PPEP MULTIOBJETIVO

Observando o fato de que “a estrutura final estável da proteína é dada naturalmente por um conjunto de estruturas semelhantes, em função da entropia presente nas conformações instáveis iniciais”, Cutello *et al.* (2006) propõem que as interações entre átomos ligados e átomos não-ligados devem ser vistas como objetivos distintos durante a otimização. Deste modo, formula-se o PPEP como um MOP (CUTELLO *et al.*, 2006):

$$\text{Minimizar } \mathbf{f}(\mathbf{x}) = \left\{ \begin{array}{l} f_{\text{ligados}}(\mathbf{x}) = E_{\text{ligacoes}} + E_{\text{angulos}} + E_{\text{diedrais}} + E_{UB} + E_{\text{improp}}, \\ f_{\text{nao-ligados}}(\mathbf{x}) = E + E_{VW} \end{array} \right\} \quad (25)$$

sujeito, para cada resíduo R da proteína, às restrições:

$$\begin{aligned} \mathbf{x} &= (x_1 = \Phi_R, x_2 = \Psi_R, x_3 = \chi_{R,1}, \dots) \\ \Phi_R^L &\leq \Phi_R \leq \Phi_R^U \\ \Psi_R^L &\leq \Psi_R \leq \Psi_R^U \\ -180 &\leq \chi_{R,1}, \dots \leq 180 \end{aligned}$$

onde \mathbf{x} são os ângulos de torção da proteína, os ângulos diedrais (Φ e Ψ), e os ângulos livres de cada tipo de resíduo ($\chi_1, \chi_2 \dots$), conforme a quantidade de ângulos do resíduo (Tabela 1), para todos os resíduos da proteína. Os limites Φ^L , Φ^U , Ψ^L e Ψ^U são listados na Tabela 2, conforme a predição da estrutura secundária de Kabsch e Sander (1983).

Tabela 1: Número de ângulos livres de cada resíduo

Resíduo	Ângulos χ
Ser, Cys, Thr, Val	χ_1
Ile, Leu, Asp, Asn, His, Phe, Tyr, Trp	χ_1, χ_2
Met, Glu, Gln	χ_1, χ_2, χ_3
Lys, Arg	$\chi_1, \chi_2, \chi_3, \chi_4$

Tabela 2: Restrições dos ângulos diedrais (Φ e Ψ) para cada estrutura secundária

Estrutura secundária	Φ^L	Φ^U	Ψ^L	Ψ^U
Hélice- α	-75	-55	-50	-30
Ponte- β	-130	-110	110	130
Folha- β	-150	-30	-100	50
Hélice-3 – 10	-230	-30	100	200
Hélice- π	30	130	130	260
Volta	30	150	-60	90
Torção	-160	-50	50	100
Desconhecida	-180	180	-180	180

Como exemplo, vejamos o caso da proteína *Met-Enkephalin* (MARCOTTE *et al.*, 2004), um peptídeo encontrado no sistema nervoso humano. Sua estrutura terciária foi determinada por Ressonância Magnética Nuclear e sua conformação estável, bem como outros detalhes são encontrados na base de proteínas *Protein Data Bank* (PDB) (BERMAN *et al.*, 2000) pelo identificador 1PLW. *Met-Enkephalin* é composta por cinco resíduos {Tyr,Gly,Gly,Phe,Met}. A Figura 9 mostra os átomos que compõem a estrutura da proteína e sua estrutura tridimensional.

De acordo com a Tabela 1, o resíduo Metionina possui três ângulos χ , Tirosina e Fenilalanina possuem dois e o resíduo Glicina não possui ângulos livres. Portanto, com os ângulos diedrais (Φ e Ψ) de cada resíduo, formam um total de 17 variáveis:

$$\mathbf{x} = \{\Phi_{Tyr}, \Psi_{Tyr}, \chi_{Tyr,1}, \chi_{Tyr,2}, \chi_{Tyr,3}, \Phi_{Gly1}, \Psi_{Gly1}, \Phi_{Gly2}, \Psi_{Gly2}, \\ \Phi_{Phe}, \Psi_{Phe}, \chi_{Phe,1}, \chi_{Phe,2}, \Phi_{Met}, \Psi_{Met}, \chi_{Met,1}, \chi_{Met,2}\}$$

Como todos os resíduos desta proteína possuem estrutura secundária desconhecida, todos os ângulos são livres e podem assumir valores entre -180 e 180 , de acordo com a Tabela 2.

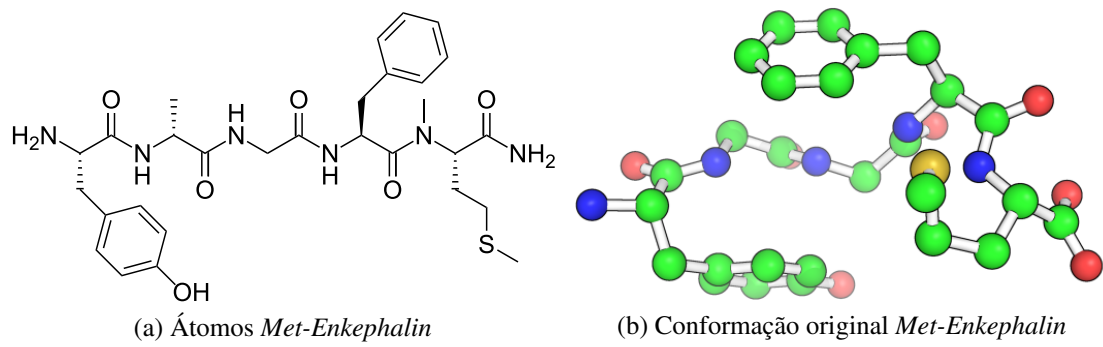


Figura 9: Estrutura da proteína *Met-Enkephalin*: (a) os átomos que compõem a proteína e (b) sua estrutura terciária estável em 3 dimensões.

3 MOEA/D COM *SURROGATES* ELM (ELMOEA/D)

O capítulo anterior fundamenta alguns conceitos sobre otimização multiobjetivo, o algoritmo MOEA/D e suas variantes MOEA/D-DE e MOEA/D-STM, a integração do MOEA/D com modelos *surrogates* no algoritmo MOEA/D-RBF, seleção automática de operadores e redes ELM para regressão. Partindo disso, este capítulo descreve a proposta da plataforma chamada de ELMOEA/D que integra o algoritmo MOEA/D com redes tipo ELM.

Primeiramente, a Seção 3.1 apresenta uma breve revisão dos trabalhos relacionados à proposta. Em seguida na Seção 3.2, a plataforma ELMOEA/D é descrita e cada passo é detalhado. Finalmente, a Seção 3.3 faz uma breve discussão sobre as diferenças da proposta com o algoritmo MOEA/D-RBF, na qual foi inspirada.

3.1 TRABALHOS CORRELATOS

Com objetivo de contextualizar a proposta frente a outras da literatura, esta seção cita e comenta brevemente alguns trabalhos correlatos. Os principais trabalhos selecionados propõem algoritmos evolutivos assistidos por *surrogate*, alguns envolvendo o algoritmo MOEA/D e outros que envolvem modelos ELM na aproximação do *fitness*. Até onde vai o conhecimento do autor, o uso de MOEA/D com *surrogates* baseados em ELM ainda não foi explorado, exceto pelo trabalho do próprio autor desta dissertação (PAVELSKI *et al.*, 2014).

3.1.1 MODELOS *SURROGATES* EM MOEAS

Foram escolhidas algumas propostas relacionadas à otimização multiobjetivo com *surrogates* como uma breve revisão da literatura relevante. Ainda outros estudos importantes relacionados ao tema poderiam ser mencionados, mas seus conceitos vão além do escopo desse trabalho. Para uma revisão mais abrangente de algoritmos evolutivos assistidos por *surrogate*, consultar os trabalhos Jin (2005) e Jin (2011).

Nain e Deb (2003) utilizam uma rede neural artificial combinada com o MOEA *Non-*

dominated Sorting Genetic Algorithm (NSGA-II). A proposta começa a busca com uma aproximação imprecisa de todo o espaço objetivo e, no decorrer da busca, a aproximação é refinada com as novas soluções. O algoritmo alterna o uso do modelo aproximado e a função original durante a evolução (controle de evolução baseado em gerações (JIN, 2005)). A proposta produz bons resultados em um problema de adequação de curvas *B-spline* com 39 variáveis e 2 objetivos.

Emmerich e Naujoks (2004) utilizam *surrogates* baseados em processos gaussianos para estimar o *fitness* e informações sobre o erro (ou variância) da aproximação. Três estratégias para selecionar novas soluções para serem avaliadas em cada geração (controle de evolução baseado em *pre-screening* (JIN, 2005)) são testadas: (1) número contante de soluções selecionadas (comparadas com os limites inferiores da aproximação, ou seja a aproximação menos o valor de erro ou variância), (2) princípio do menor erro (soluções cujos limites inferiores são dominados pelos limites superiores de outras são descartadas) e (3) princípio da melhor qualidade (médias das aproximações são comparadas). O modelo é hibridizado ao esquema de seleção dos algoritmos ϵ -MOEA e NSGA-II. A variante com NSGA-II e *pre-screening* com número contante de soluções selecionadas mostrou melhores resultados em relação aos demais para um problema *benchmark* de 20 variáveis e 2 objetivos e no problema de design aerodinâmico de asas com 18 variáveis e 2 objetivos.

Knowles e Hughes (2005) propõem o algoritmo ParEGO, uma adaptação do algoritmo *Efficient Global Optimization* (JONES *et al.*, 1998) para otimização multiobjetivo. Portanto, ParEGO utiliza a parametrização de processos gaussianos tipo DACE (*design and analysis of computer experiments*) para estimar a verossimilhança máxima dos objetivos. As soluções do arquivo de soluções avaliadas que minimizam a função de agregação Tchebycheff (TCHE), dado um vetor de pesos aleatório, são utilizadas para treinar o modelo em cada iteração. Após isso, um algoritmo genético *steady-state* encontra a melhor solução para ser avaliada e incluída no arquivo, evoluindo uma população com os objetivos estimados pelo modelo *surrogate*. O algoritmo é comparado ao NSGA-II e apresenta melhores resultados na maior parte dos nove *benchmarks* propostos, com 2 a 8 variáveis de decisão e 2 ou 3 objetivos.

Rosales-Perez *et al.* (2013) propõem um MOEA com um *ensemble* de máquinas de vetores suporte (SVMs, do inglês *Support Vector Machines*) para aproximar a função de *fitness*. A proposta utiliza arquivos de soluções para armazenar conhecimento durante a busca e treinar os *surrogates* quando a aproximação torna-se pior que uma solução gerada aleatoriamente. Inicialmente, os parâmetros para SVM são determinados a partir do erro de aproximação na população inicial e, por um número de gerações, novas soluções são geradas com operadores genéticos, as soluções são então avaliadas com os *surrogates* a fim de encontrar soluções

candidatas para avaliação. As soluções não-dominadas são avaliadas com a função original custosa, adicionadas aos arquivos e utilizadas para gerar novos modelos *surrogates*. O *ensemble* é construído pelos *surrogates* gerados até um limite de gerações anteriores (fila de modelos). A proposta foi comparada ao NSGA-II para os problemas *benchmark* ZDT (ZITZLER; THIELE, 1998) com 10 a 30 variáveis e 2 objetivos e obteve resultados inferiores apenas em um problema com frentes de Pareto locais (ZDT4). Em um trabalho mais recente (ROSALES-PEREZ *et al.*, 2015), a proposta também obteve sucesso ao reduzir a número de avaliações em um MOP envolvendo a seleção de parâmetros para classificadores, com 12 variáveis e 2 objetivos.

3.1.2 MODELOS *SURROGATES* COM O ALGORITMO MOEA/D

O algoritmo MOEA/D já foi considerado para otimização multiobjetivo com poucas avaliações. De acordo com o conhecimento dos autores, apenas dois trabalhos exploram o algoritmo neste contexto.

Zhang *et al.* (2010) propõem uma adaptação do MOEA/D, MOEA/D-EGO, para MOPs custosos, onde processos gaussianos e *Fuzzy C-means* para estimar a “melhora esperada” de cada solução de maneira local. Assim como no ParEGO, a proposta busca explorar regiões promissoras baseando-se na média e variância dos modelos. O MOEA/D-EGO é testado nos problemas ZDT com 8 variáveis e outros MOPs *benchmarks* de 2 a 6 variáveis e mostrou não ser pior que o ParEGO e SMS-EGO (PONWEISER *et al.*, 2008). O SMS-EGO também é um MOEA baseado no algoritmo ParEGO e processos gaussianos, porém o modelo serve para estimar as contribuições de cada solução para o Hipervolume da população.

Martínez e Coello (2013) apontam que processos gaussianos possuem alto custo computacional, mesmo para um número médio de variáveis e propõem o MOEA/D com redes neurais tipo RBF (MOEA/D-RBF), já mencionado na Seção 2.3.3. O MOEA/D-RBF mostrou ser superior ao MOEA/D-EGO na maioria dos problemas ZDT com 8 variáveis e 2 objetivos, apresenta bons resultados nos mesmos problemas com o número original de variáveis (30 e 10 variáveis) e no problema de design aerodinâmico de asas com 12 variáveis e 2 objetivos.

3.1.3 ELM COMO MODELO *SURROGATE*

Muitos algoritmos propostos permitem que qualquer modelo de regressão seja adotado como *surrogate*, porém, resultados empíricos de ELMs nesse contexto são escassos. Apenas os dois trabalhos seguintes são do conhecimento do autor desta dissertação.

Lim *et al.* (2007) comparam quatro modelos *surrogate* diferentes: processos gaussianos,

regressão polinomial, redes RBF e ELM. A versão não regularizada do treinamento da ELM é utilizada, ou seja, os pesos são determinados multiplicando a pseudo-inversa da matriz de saídas dos neurônios ocultos pelo vetor de saídas, assim como na Equação 21. Os modelos *surrogate* são utilizados na busca local de um algoritmo memético mono-objetivo. Processos gaussianos e regressão polinomial foram as melhores estratégias em quatro problemas de otimização mono-objetivo de 10 variáveis.

Hao e Liu (2014) utilizam *surrogates* baseados em uma versão incremental da ELM (GUO *et al.*, 2014) junto ao algoritmo DE para estimar o objetivo em problemas discretos. A cada iteração, os 10% dos indivíduos gerados, com melhor *fitness*, são avaliados pela função custosa (controle de evolução *pre-screening* (JIN, 2005)). O algoritmo é comparado com uma variante do algoritmo DE não-baseada em *surrogate* e é superior na maioria das instâncias de um problema de alocação mono-objetivo com até 500 variáveis binárias.

3.2 ELMOEA/D: ALGORITMO MULTIOBJETIVO BASEADO EM DECOMPOSIÇÃO E SURROGATES ELM

Como mencionado, MOEAs baseados em *surrogate* utilizam modelos computacionalmente baratos para estimar o *fitness* e reduzir o número de avaliações de funções-objetivo custosas. Esta seção trata da proposta de uma plataforma para integrar o algoritmo MOEA/D e duas de suas variantes com modelos de regressão ELM, chamando-a ELMOEA/D. A plataforma proposta, cujo fluxo de execução está ilustrado na Figura 10, é inspirada no algoritmo MOEA/D-RBF (MARTÍNEZ; COELLO, 2013), descrito na Seção 2.3.3.

O diagrama na Figura 10 ilustra cada fase executada pela plataforma proposta, o fluxo de execução e os arquivos de soluções utilizados, passos que serão descritos em mais detalhes no Algoritmo 5 e nas seções seguintes. O algoritmo ELMOEA/D mantém um arquivo P_E com as soluções avaliadas pela função custosa do problema e um arquivo P_T com soluções selecionadas para formar o conjunto de treinamento dos modelos *surrogate*.

Conforme ilustrado no Algoritmo 5, o procedimento do ELMOEA/D consiste em: (1) gerar soluções iniciais e adicioná-las aos arquivos; (2) treinar os modelos *surrogate* com base no arquivo de treinamento; (3) gerar a população da primeira geração do MOEA/D com base nas soluções avaliadas e otimizar as funções-objetivo aproximadas pelos modelos *surrogate*; (4) selecionar quais soluções serão avaliadas com a função custosa; (5) atualizar os arquivos P_E e P_T e, finalmente, caso o número máximo de avaliações seja alcançado, retornar as soluções não-dominadas. Os passos que compõem o Algoritmo 5 serão detalhados ao longo das próximas seções.

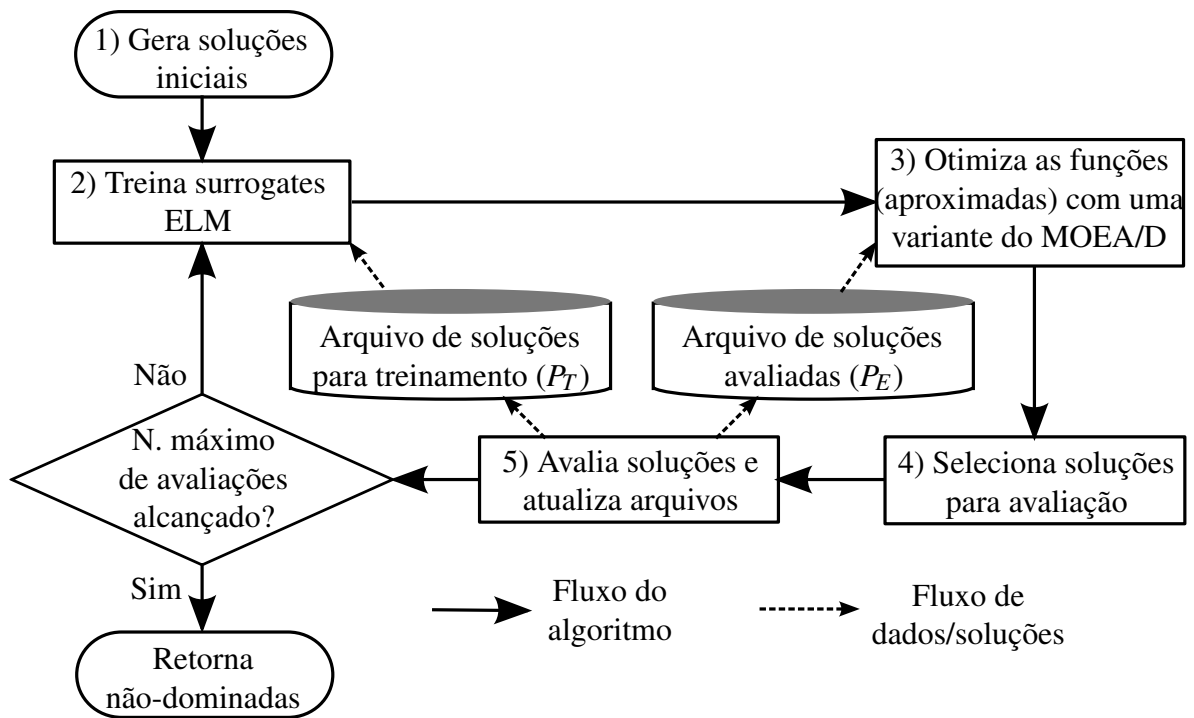


Figura 10: Fluxo de execução da plataforma do ELMOEA/D

Algoritmo 5 ELMOEA/D

Entrada: MOP com função-objetivo custosa \mathbf{f} a ser otimizada e o número de avaliações N_E ;
 W : conjunto de N vetores de pesos para o algoritmo MOEA/D;
 W_T : conjunto de N_T vetores de pesos para a atualização do arquivo de treinamento;
 W_S : conjunto de N_S vetores de pesos para a seleção de soluções para avaliação;
 Demais parâmetros dos algoritmos MOEA/D e ELM.

Saída: Aproximação do conjunto de Pareto (\mathcal{APS})

- 1: $P_T^0 \leftarrow P_E^0 \leftarrow GeracaoSolucoesIniciais(\mathbf{f}, \mathbf{x}_L, \mathbf{x}_U)$ // Seção 3.2.1
 - 2: $c \leftarrow N_T, g \leftarrow 0$
 - 3: **enquanto** $c < N_E$ **faça**
 - 4: $\mathbf{f}_a \leftarrow TreinamentoSurrogates(P_T^g)$ // Seção 3.2.2
 - 5: $P^0 \leftarrow InicializacaoPopulacaoMOEAD(NaoDominadas(P_E^g), \mathbf{f}_a)$ // Seção 3.2.3
 - 6: $P^G \leftarrow EvolucaoMOEAD(\mathbf{f}_a, P^0, W)$
 - 7: $P_S^g \leftarrow SelecaoSolucoes(P^G, W, W_S, P_E^g)$ // Seção 3.2.4
 - 8: Avalia as soluções de P_S^g com a função custosa \mathbf{f}
 - 9: $P_E^{g+1} \leftarrow P_E^g \cup P_S^g$ // Seção 3.2.5
 - 10: $P_T^{g+1} \leftarrow AtualizacaoArquivoTreinamento(P_T^g, W_T, P_S^g)$
 - 11: $c \leftarrow c + |P_S^g|, g \leftarrow g + 1$
 - 12: **fim enquanto**
 - 13: **retorne** Soluções não-dominadas de P_E^g .
-

3.2.1 GERAÇÃO DE SOLUÇÕES INICIAIS

Primeiramente, um conjunto de N_T soluções é gerado. As soluções podem ser conhecidas previamente ou geradas aleatoriamente dentro dos limites do problema (\mathbf{x}_L e \mathbf{x}_U). A

amostragem estratificada *Latin Hypercube Sampling* (LHS) (MCKAY *et al.*, 1979), notável por gerar soluções bem espaçadas no espaço de busca, é utilizada.

Segundo o procedimento da amostragem LHS, cada dimensão do espaço de decisão é dividida em N_T segmentos (número de soluções). Cada segmento é aleatoriamente designado sem repetições a uma única solução. Em cada solução, seleciona-se um valor aleatório para cada variável de decisão, dentro dos limites de seus segmentos sorteados.

As soluções geradas são avaliadas pela função custosa e incluídas no arquivo de treinamento (P_T^0) e no arquivo de soluções avaliadas (P_E^0). Até que o limite máximo de N_E avaliações seja alcançado, o algoritmo itera nos passos descritos nas seções seguintes.

3.2.2 TREINAMENTO DO *SURROGATE* ELM

O arquivo de treinamento contém um número fixo de N_T soluções avaliadas com a função original custosa. Os valores das variáveis de decisão e objetivos destas soluções serão utilizados como conjunto de treinamento para uma técnica de regressão supervisionada, neste caso, redes do tipo ELM. Como visto na Seção 2.5, o treinamento da ELM envolve selecionar valores aleatórios para os parâmetros dos neurônios ocultos ($\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{N_H}$ e b_1, b_2, \dots, b_{N_H}) e calcular os pesos de saída ($\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_{N_O}$) com a Equação 22 ou 23. Assim como descrito na Equação 17, a aproximação do MOP custoso \mathbf{f} (formalizado na Equação 1) pela rede ELM com N_O saídas será dada por:

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimizar}} \quad & \mathbf{f}_a(\mathbf{x}) = \left\{ f_{a,1}(\mathbf{x}), \dots, f_{a,N_O}(\mathbf{x}) : f_{a,j}(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta}_j \right\} \\ \text{sujeito a} \quad & \mathbf{x} \in \Omega \text{ ou seja, } x_d^L \leq x_d \leq x_d^U \quad d = 1, 2, \dots, N_D \end{aligned} \quad (26)$$

Como as saídas dos neurônios ocultos ($\mathbf{h}(\mathbf{x})$) são as mesmas em todas as funções, apenas os pesos de saída ($\boldsymbol{\beta}_0, \dots, \boldsymbol{\beta}_{N_O}$) são ajustados, conforme a Equação 22 ou 23. Ainda, a inversão de matriz necessária nas Equações 22 e 23 pode ser substituída por um sistema linear em que a mesma fatoração da matriz de coeficientes é utilizada no ajuste dos pesos das múltiplas saídas. Por exemplo, o cálculo dos pesos de saída na Equação 22 pode ser substituído pelo sistema linear:

$$H' \boldsymbol{\beta} = H^T \mathbf{v} \quad (29)$$

$$H' = \left(H^T H + \frac{I}{C} \right) \quad (30)$$

Como a matriz H' é definida positiva simétrica, ela pode ser fatorada em $H' = LL^T$, onde L é uma matriz triangular inferior. Este procedimento é conhecido como fatoração de Cholesky e,

nesse caso, possui complexidade $O(N_H^3)$ (DEMMELE, 1997). A resolução dos dois sistemas lineares resultantes:

$$L\boldsymbol{\beta}' = H^T \mathbf{v} \quad (33)$$

$$L^T \boldsymbol{\beta} = \boldsymbol{\beta}' \quad (34)$$

por substituição ascendente e descendente pode ser feita em $O(N_H^2)$ operações. O mesmo vale para a Equação 23 e, utilizando a mesma decomposição $H' = LL^T$ para N_O saídas, a complexidade do ajuste de pesos da ELM será $O(\min(N_H, N_T)^3 + N_O \min(N_H, N_T)^2)$.

Portanto, o treinamento do *surrogate* ELM pode ser realizado de maneira eficiente, utilizando implementações otimizadas para resolução de sistemas lineares, e possui boa escalabilidade em relação ao número de objetivos.

3.2.3 POPULAÇÃO INICIAL E OTIMIZAÇÃO DA FUNÇÃO APROXIMADA COM MOEA/D

O conhecimento das melhores soluções até o momento é útil para iniciar ou continuar a busca por soluções promissoras a serem avaliadas com a função original \mathbf{f} . Desse modo, a população inicial do MOEA/D (P^0) é gerada a partir das soluções não-dominadas do arquivo de soluções avaliadas P_E . Ao incluir as N_{nd} soluções não-dominadas na população inicial do MOEA/D de tamanho N , obtemos três situações possíveis:

- $N_{nd} = N$ e todas são adicionadas à população inicial;
- $N_{nd} > N$ e N soluções não-dominadas são selecionadas aleatoriamente para compor P^0 ;
- $N_{nd} < N$, nesse caso as N_{nd} soluções são adicionadas à população, e o restante é gerado com o LHS nos limites $[\boldsymbol{\mu} - \boldsymbol{\sigma}, \boldsymbol{\mu} + \boldsymbol{\sigma}]$, onde $\boldsymbol{\mu}$ e $\boldsymbol{\sigma}$ são a média e desvio padrão das variáveis de decisão das N_{nd} soluções não-dominadas. As novas soluções, como não estão presentes em P_E , são avaliadas pela função aproximada \mathbf{f}_a (Equação 17).

Após inicializar a população, alguma variante do algoritmo MOEA/D, descrita na Seção 2.3, é utilizada para encontrar N soluções que buscam minimizar as funções aproximadas \mathbf{f}_a no MOP formalizado na Equação 26. Deste modo, quando um novo indivíduo é gerado, a avaliação dos objetivos é feita pela aproximação \mathbf{f}_a , composta por modelos de regressão ELM de baixo custo computacional. Dados o conjunto de pesos \mathbf{W} e os parâmetros pertinentes (tamanho da vizinhança N_B , número máximo de atualizações N_R , etc), o MOEA/D evolui a população P^0 por G gerações.

3.2.4 SELEÇÃO DE SOLUÇÕES PARA AVALIAÇÃO

A população final do MOEA/D, P^G , será constituída de N soluções que buscam minimizar f_a e podem também minimizar a função original f (vide Figura 2). Nesta fase, N_S soluções serão selecionadas, preferencialmente próximas e bem distribuídas ao longo da aproximação da frente de Pareto gerada pela população P^G . Assim, a seguinte estratégia de seleção é adotada:

- dado um conjunto de vetores de pesos no espaço de objetivos W_S , onde $|W_S| = N_S$;
- para cada vetor de peso $w_S \in W_S$, a associação $B_S(w_S) = \{w_{n1}, w_{n2}, \dots\}$ é definida como contendo os $\lfloor N/N_S \rfloor$ vetores de pesos do algoritmo MOEA/D mais próximos de acordo com a distância euclidiana, como ilustrado na Figura 11;

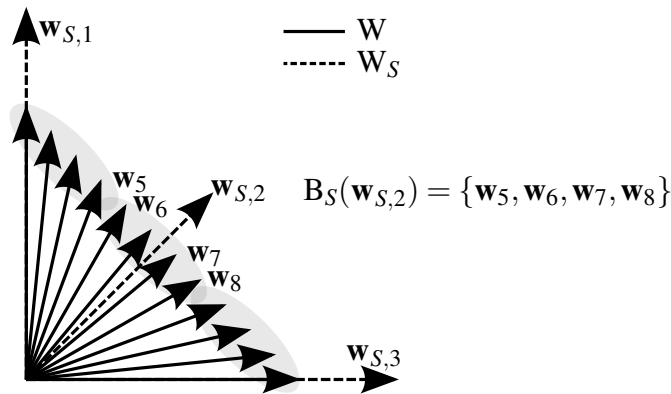


Figura 11: Associação entre os pesos do MOEA/D (W) e os pesos utilizados na seleção (W_S).

- para cada conjunto de pesos $B_S(w_S)$, as soluções são ordenadas de acordo com a função de agregação $F(\cdot)$, a mesma utilizada pelo algoritmo MOEA/D;
- caso a solução não seja similar a nenhuma outra em P_E , ela é selecionada para avaliação e marcada como “vista”. Se a melhor solução não vista de cada conjunto $B_S(w_S)$ for similar a alguma contida no arquivo de soluções avaliadas P_E , a solução é marcada como “vista” e o próximo conjunto é verificado;
- os conjuntos são varridos até que (1) N_S soluções sejam selecionadas, (2) todas as soluções da população P^G tenham sido vistas ou (3) o número máximo de avaliações tenha sido alcançado;

Uma solução x é considerada similar à solução y se a distância entre as duas soluções é menor que ε no espaço de decisão, i.e., x é similar a y se $\|x - y\| < \varepsilon$. Se o total de soluções selecionadas para avaliação for menor do que $0.1N_S$, a distância mínima é reduzida pela metade ($\varepsilon \leftarrow \varepsilon/2$) e a seleção é repetida.

Finalmente, as soluções selecionadas neste passo são avaliadas com a função original custosa \mathbf{f} e compõem o arquivo P_S^g .

3.2.5 ATUALIZAÇÃO DOS ARQUIVOS

As novas soluções avaliadas (P_S^g) são adicionadas aos arquivos de soluções avaliadas e de treinamento. O arquivo de soluções avaliadas não possui restrições de tamanho, portanto todas as soluções são incluídas. Cada solução adicionada serve também para atualizar a informação de quais soluções no arquivo são não-dominadas. Para cada solução do arquivo P_E marcada como não-dominada (\mathbf{x}_{nd}), verifica-se, para a nova solução \mathbf{x}^* :

- se $\mathbf{x}_{nd} \prec \mathbf{x}^*$, a nova solução \mathbf{x}^* é marcada como dominada e a verificação pára;
- se $\mathbf{x}^* \prec \mathbf{x}_{nd}$, a solução do arquivo \mathbf{x}_{nd} é marcada como dominada e a verificação continua com a próxima solução não-dominada do arquivo;
- caso \mathbf{x}^* e \mathbf{x}_{nd} sejam incomparáveis ($\mathbf{x}_{nd} \not\prec \mathbf{x}^*$ e $\mathbf{x}^* \not\prec \mathbf{x}_{nd}$), a verificação continua com a próxima solução não-dominada do arquivo e, se não houver outra solução a ser comparada, \mathbf{x}^* é marcada como não-dominada.

Este processo evita que a ordenação por não-dominância seja realizada toda vez que as soluções não-dominadas do arquivo são necessárias, como na geração da população inicial do MOEA/D e no passo final para retornar todas as soluções não-dominadas encontradas.

Ao contrário do arquivo de soluções avaliadas (P_E), o arquivo de treinamento (P_T) possui tamanho limitado de N_T soluções e a nova solução deve substituir uma solução presente. Toda nova solução avaliada, mesmo que dominada por outras, é adicionada ao arquivo de treinamento, o que pode ajudar caso a busca guiada pela ELM esteja na direção errada. Dado o conjunto W_T , com N_T vetores de pesos distribuídos uniformemente no espaço de objetivos, a solução substituída pela nova solução \mathbf{x}^* será:

- \mathbf{x}_k , associada ao vetor de pesos $\mathbf{w}_{T,k}$, onde $\mathbf{w}_{T,k}$ é o vetor de pesos de W_T em que \mathbf{x}^* obtém o menor valor de função de agregação e, ao mesmo tempo, supera ou equivale à solução \mathbf{x}_k . Desse modo:

$$\mathbf{w}_{T,k} = \min_{\mathbf{w}_T} \{F(\mathbf{x}^*, \mathbf{w}_T, \mathbf{z}) : \mathbf{w}_T \in W_T\}$$

e $F(\mathbf{x}^*, \mathbf{w}_{T,k}, \mathbf{z}) \leq F(\mathbf{x}_k, \mathbf{w}_{T,k}, \mathbf{z})$. A mesma função de agregação $F(\cdot)$ do algoritmo MOEA/D é utilizada neste passo;

- caso o valor da função de agregação para \mathbf{x}^* seja maior ($F(\mathbf{x}^*, \mathbf{w}_{T,k}, \mathbf{z}) > F(\mathbf{x}_k, \mathbf{w}_{T,k}, \mathbf{z})$), ou seja, a nova solução \mathbf{x}^* não melhoraria a solução \mathbf{x}_k em termos de função de agregação, a solução mais próxima no espaço de objetivos é substituída. Portanto a nova solução \mathbf{x}^* sempre será incluída e substituirá a solução:

$$\mathbf{x}^k = \min_{\mathbf{x}} \{ \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}^*)\| : \mathbf{x} \in P_T \}.$$

3.2.6 MÚLTIPLOS *SURROGATES*

O gerenciamento de múltiplos modelos *surrogates* pode reduzir o erro e tornar a aproximação do *fitness* mais robusta (JIN, 2011). O proposto para a utilização de múltiplos modelos *surrogate* é que o espaço de objetivos seja dividido, de modo que cada ELM será associada a um vetor de pesos, será treinada e estimará soluções cujos vetores de pesos estão próximos ao vetor de pesos da ELM, assim como a associação entre soluções vizinhas e a seleção de soluções candidatas do ELMOEA/D (Seção 3.2.4).

Dado um conjunto de vetores de pesos $\mathbf{W}_a = \{\mathbf{w}_{a,1}, \dots, \mathbf{w}_{a,N_a}\}$, espalhados uniformemente no espaço de objetivos, com tamanho igual ao número de *surrogates* especificado N_a , tal que N seja múltiplo de N_a . Durante a fase de treinamento (Seção 3.2.2), cada vetor de peso \mathbf{w}_a será associado a $N_{a,T}$ vetores de pesos de \mathbf{W}_T :

$$\mathbf{B}_{a,T}(\mathbf{w}_{a,k}) = \{ \mathbf{w}_{T,n1}, \dots, \mathbf{w}_{T,nN_{a,T}} \}$$

onde $\mathbf{w}_{T,n1}, \dots, \mathbf{w}_{T,nN_{a,T}}$ são os $N_{a,T}$ vetores de pesos mais próximos a $\mathbf{w}_{a,k}$. Assim, ao invés de usar todas as soluções do arquivo de treinamento em um único *surrogate*, cada um é treinado individual e localmente. As $N_{a,T}$ soluções associadas aos vetores de pesos em $\mathbf{B}_{a,T}(\mathbf{w}_{a,k})$, $\mathbf{x}_{n1}, \dots, \mathbf{x}_{nN_{a,T}}$, serão utilizadas para treinar o k -ésimo *surrogate* referente a $\mathbf{w}_{a,k}$.

Consequentemente, ao estimar uma solução $\mathbf{x}_i \in P$, durante a fase de otimização da função aproximada (Seção 3.2.3), deseja-se escolher o modelo *surrogate* mais apropriado à sua direção de busca ou vetor de pesos referente. Com isso, o modelo que estimará \mathbf{x}_i será o modelo k cujo conjunto de pesos do MOEA/D associado inclui \mathbf{w}_i , i.e., $\mathbf{w}_i \in \mathbf{B}_{a,E}(\mathbf{w}_{a,k})$, onde $\mathbf{B}_{a,E}(\mathbf{w}_{a,k})$ é dado por:

$$\mathbf{B}_{a,E}(\mathbf{w}_{a,k}) = \{ \mathbf{w}_{k0}, \dots, \mathbf{w}_{k\lfloor N/N_a \rfloor} \}$$

tal que $\mathbf{B}_{a,E}(\mathbf{w}_{a,k})$ serão os $\lfloor N/N_a \rfloor$ pesos de \mathbf{W} mais próximos a $\mathbf{w}_{a,k}$.

Nota-se que o número de soluções utilizadas no treinamento de cada *surrogate* é dado pelo parâmetro $N_{a,T}$ e, desse modo, pode haver soluções que participam do treinamento de mais

de um *surrogate* (caso $N_{a,T} > \lfloor N_T/N_a \rfloor$). Porém, durante a avaliação de uma solução, esta só poderá ser avaliada por exatamente um modelo *surrogate*, desse modo o número de soluções N deve ser divisível pelo número de *surrogates* N_a .

3.3 DIFERENÇAS COM O ALGORITMO MOEA/D-RBF

O algoritmo MOEA/D-RBF foi utilizado como inspiração para compor o ELMOEA/D proposto. As principais diferenças entre o os algoritmos ELMOEA/D e MOEA/D-RBF são:

Uso de diferentes variantes MOEA/D: o MOEA/D-RBF foi testado unicamente com o algoritmo MOEA/D original (Algoritmo 1) com operadores genéticos. Na proposta do ELMOEA/D, a estratégia utilizada para realizar a busca com as funções-objetivo aproximadas não se limita a uma variante do MOEA/D, ela inclui, além do algoritmo original, as variantes MOEA/D-DE e MOEA/D-STM;

Uso de ELM como *surrogate*: como mencionado na Seção 2.3.3, o algoritmo MOEA/D-RBF é assistido por um *ensemble* de 3 redes tipo RBF, treinadas a partir da clusterização das entradas e otimização dos pesos de saída pelo algoritmo DE. Por sua vez, o algoritmo ELMOEA/D tem como princípio a utilização de uma rede tipo ELM, que possui maior simplicidade, eficiência computacional e não exige muitos parâmetros (função de ativação, N_H e C);

Múltiplas ELMs como *surrogate*: a Seção 3.2.6 propõe que várias ELMs sejam utilizadas na aproximação da função-objetivo. Neste caso, cada ELM é treinada com soluções referentes a uma partição do espaço de objetivos. O algoritmo MOEA/D-RBF não faz uso desta estratégia, pois a partição é dirigida pela estratégia de cooperação das redes RBF e feita no espaço das variáveis de decisão;

Critério de similaridade: durante a escolha da solução a ser avaliada, a nova solução é considerada diferente de outra já avaliada se a distância entre as duas for menor que o valor do parâmetro ε . Na proposta do algoritmo MOEA/D-RBF este critério não é explicitado;

4 EXPERIMENTOS E RESULTADOS

Tendo sido descrita a plataforma ELMOEA/D proposta neste trabalho, este capítulo pretende avaliar seu desempenho em diferentes cenários. Todos os experimentos foram realizados em um computador Intel Core I5 com 4Gb de memória RAM e sistema operacional Linux. Os algoritmos foram implementados na linguagem C++ com o compilador GCC versão 4.9 e otimizações nível 2 (*flag -O2*).

Na primeira etapa (Seção 4.2), os algoritmos MOEA/D original, MOEA/D-DE e MOEA/D-STM (descritos nas Seções 2.3, 2.3.1 e 2.3.2) são incorporados, em diferentes momentos, à plataforma ELMOEA/D para avaliar o uso destas variantes do MOEA/D como otimizador. Em todos os experimentos, o otimizador é assistido por ELMs como modelos *surrogate*. A melhor variante (ELMOEA/D com MOEA/D original) é comparada com o algoritmo MOEA/D-RBF, que inspirou o ELMOEA/D, porém utiliza redes RBF como modelo *surrogate*. Como base de comparação, os resultados do algoritmo MOEA/D original, sem *surrogates*, também são analisados.

Com base no desempenho satisfatório do algoritmo ELMOEA/D na etapa anterior, um segundo conjunto de experimentos foi realizado para aplicação da proposta ao PPEP multiobjetivo (descrito na Seção 2.6). Como a configuração utilizada nos *benchmarks* da fase anterior não produz bons resultados para o PPEP, os parâmetros do ELMOEA/D são ajustados para o PPEP de três proteínas pequenas (1PLW, 2MLT e 1ZDD). Em seguida, o ELMOEA/D com os melhores parâmetros é comparado com o algoritmo MOEA/D-DE sem o uso de *surrogates* para o PPEP com todas as proteínas testadas (1PLW, 2MLT, 1ZDD, 1CRN, 1ROP, 1UTG e 1CTF). Por fim, são apresentados os resultados da proposta em uma formulação alternativa do PPEP, com sete objetivos. O cálculo das energias para avaliação das funções-objetivo na Equação 25 é realizado pelo software para design de moléculas Tinker (PONDER, 2004), juntamente com o campo de força CHARMM v.27 (MACKERELL *et al.*, 2000).

4.1 METODOLOGIA DE AVALIAÇÃO E COMPARAÇÃO

A metodologia para comparar diferentes algoritmos e configurações em cada instância de problema segue os seguintes passos:

- gerar 30 execuções independentes de cada algoritmo;
- filtrar as soluções não-dominadas encontradas em cada execução;
- avaliar as soluções não-dominadas com os indicadores de qualidade e métricas pertinentes, gerando 30 valores de qualidade para cada algoritmo;
- interpretar as qualidades a partir da média, variância, *boxplots* e funções de *attainment* (LÓPEZ-IBÁÑEZ *et al.*, 2010) testes estatísticos.

Nos testes com *benchmarks*, utiliza-se o indicador de qualidade Hipervolume ($I_H(P, \mathbf{r})$), definido como a área dominada pelas soluções da população encontrada no espaço de objetivos (P) e até um ponto de referência (\mathbf{r}). O ponto de referência, portanto, é específico do problema e deve ser dominado pelos pontos relevantes no espaço objetivo. O indicador Hipervolume não necessita de conhecimento da frente de Pareto. Para que o valor seja minimizado (compatível com a metodologia de Martínez e Coello (2013) e Zhang *et al.* (2010)) e assuma valores pequenos (melhor visualização e precisão computacional), a métrica Hipervolume é calculada com:

$$I_H^-(P, \mathbf{r}) = I_H(P^*, \mathbf{r}) - I_H(P, \mathbf{r}) \quad (35)$$

onde P^* é a população de referência. Portanto, quanto menor o valor de $I_H^-(P, \mathbf{r})$, melhor é a população P , sendo que o indicador de qualidade Hipervolume avalia tanto a convergência quanto a diversidade (ZITZLER; KÜNZLI, 2004). Para os problemas *benchmark*, a população de referência é pré-definida, formada por 500 pontos igualmente distribuídos sobre a \mathcal{PF} (Figura 12), e o ponto de referência \mathbf{r} é específico de cada problema (Tabela 4).

Nas instâncias do PPEP, a métrica utilizada para avaliar as aproximações da \mathcal{PF} é a menor soma dos objetivos para os indivíduos não-dominados:

$$DM(P) = \min \left\{ \sum_{j=1}^{N_o} f_j(\mathbf{x}) : \mathbf{x} \in P \right\}. \quad (36)$$

A métrica Hipervolume ($I_H(P, \mathbf{r})$ ou $I_H^-(P, \mathbf{r})$) não foi considerada para o problema PPEP por três motivos: (1) a utilização da métrica com o ponto nadir como ponto de referência (ponto com os maiores objetivos de todas as aproximações, como visto em Bleuler *et al.* (2003)),

considerando $I_H(P, \mathbf{r})$ ao invés de $I_H^-(P, \mathbf{r})$, levaria a problemas de precisão numérica para números grandes de ponto flutuante, pois os valores de energia podem variar de -50 até 10^{20} ; (2) o segundo objetivo (energia dos termos não-ligados) é muito maior que o primeiro (energia dos termos ligados), portanto uma pequena contribuição no primeiro objetivo levaria a um grande acréscimo do hipervolume (área total) e os algoritmos seriam discriminados apenas de acordo com o primeiro objetivo e (3) para o problema PPEP a menor soma dos objetivos corresponde à energia total (função-objetivo original CHARMM da Equação 24). Métricas mais específicas como o RMSD (TRAMONTANO, 2006) serão consideradas em trabalhos futuros.

Após obter os valores de qualidade, o teste estatístico Kruskal-Wallis (CONOVER, 1999) auxilia na comparação entre os algoritmos. O teste de Kruskal-Wallis pode ser visto como a versão não-paramétrica do teste ANOVA, ou seja, não depende da distribuição dos valores. O teste pode ser utilizado para comparar múltiplos conjuntos de valores, desde que os valores sejam comparáveis. O teste baseia-se na comparação dos postos (ou *ranks*) dos valores observados e a hipótese nula ocorre quando as médias dos postos são semelhantes (CONOVER, 1999). Neste trabalho, um nível de significância de 0,05 foi considerado em todos os casos.

4.2 PARTE I: PROBLEMAS *BENCHMARK*

Nesta seção, comparam-se diferentes variantes do algoritmo MOEA/D na otimização da função aproximada (\mathbf{f}_a) na plataforma ELMOEA/D. Após isso, a melhor versão é comparada com o algoritmo MOEA/D-RBF. Cada configuração foi testada em cinco problemas *benchmarks* contínuos com dois objetivos, e diferentes propriedades, conhecidos como ZDT1, ZDT2, ZDT3, ZDT4 e ZDT6 (ZITZLER; THIELE, 1998).

ZDT1 possui \mathcal{PF} na forma convexa, ZDT2 na forma côncava e ZDT3 possui \mathcal{PF} desconexa; ZDT4 por sua vez, contém várias frentes locais sub-ótimas e ZDT6 é caracterizado pela distribuição não-uniforme de soluções não-dominadas no espaço de decisão (ZITZLER; THIELE, 1998). A Figura 12 apresenta o formato da \mathcal{PF} para cada problema ZDT através de 500 pontos uniformemente distribuídos sobre a \mathcal{PF} . Cada problema ZDT pode ser expresso como:

$$\begin{aligned} \text{Minimizar}_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) = \{f_1(x_1), f_2(\mathbf{x})\} \\ \text{sujeito a} \quad & f_2(\mathbf{x}) = f_2'(x_2, \dots, x_{N_D}) f_2''(f_1(x_1), f_2'(x_2, \dots, x_{N_D})) \\ & x_d^U \leq x_d \leq x_d^L \quad d = 1, \dots, N_D \end{aligned} \quad (37)$$

onde a formulação das funções f_1 , f_2' e f_2'' , bem como os limites \mathbf{x}_U e \mathbf{x}_L , são apresentadas para

cada problema na Tabela 3.

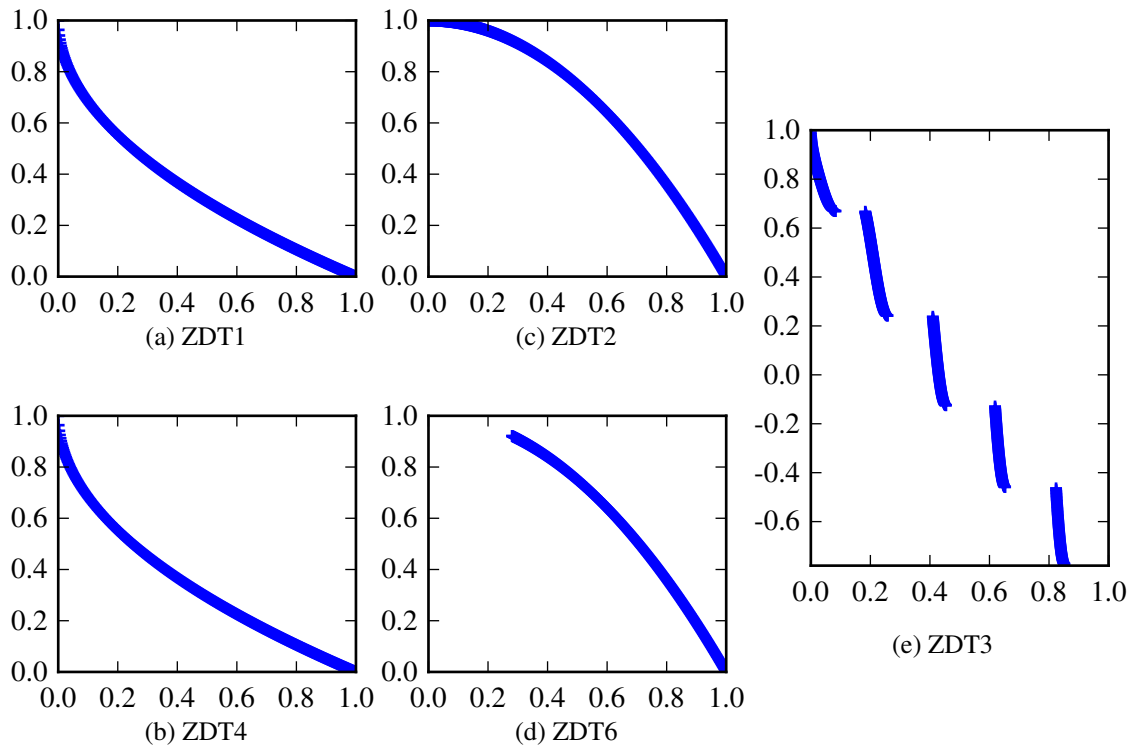


Figura 12: Frente de Pareto dos problemas *benchmark* ZDT.

Tabela 3: Formulação dos problemas ZDT: fórmulas que compõem cada problema ZDT e os limites inferiores e superiores, de acordo com a Equação 37.

MOP	$f_1(x_1)$	$f'_2(x_2, \dots, x_{N_D})$	$f''_2(f_1, f'_2)$	\mathbf{x}_L e \mathbf{x}_U
ZDT1	x_1	$1 + 9(\sum_{i=2}^{N_D} x_i / (N_D - 1))$	$1 - \sqrt{f_1/f'_2}$	$\mathbf{x} \in [0, 1]$
ZDT2	x_1	$1 + 9(\sum_{i=2}^{N_D} x_i / (N_D - 1))$	$1 - (f_1/f'_2)^2$	$\mathbf{x} \in [0, 1]$
ZDT3	x_1	$1 + 9(\sum_{i=2}^{N_D} x_i / (N_D - 1))$	$1 - \sqrt{f_1/f'_2} - (f_1/f'_2) \text{sen}(10\pi f_1)$	$\mathbf{x} \in [0, 1]$
ZDT4	x_1	$1 + 10(N_D - 1) + \sum_{i=2}^{N_D} (x_i^2 - 10 \cos(4\pi x_i))$	$1 - \sqrt{f_1/f'_2}$	$\mathbf{x} \in [-5, 5],$ $x_1 \in [0, 1]$
ZDT6	$1 - e^{-4x_1} \text{sen}^6(6\pi x_1)$	$1 + 9(\sum_{i=2}^{N_D} x_i / (N_D - 1))^{0.25}$	$1 - (f_1/f'_2)$	$\mathbf{x} \in [0, 1]$

O número de variáveis é ajustável e os autores sugerem $N_D = 30$ para ZDT1-3 e $N_D = 10$ para ZDT4 e ZDT6 (ZITZLER; THIELE, 1998). Neste caso, estes problemas foram analisados com um número reduzido de $N_E = 1000$ avaliações, sendo que, na literatura, os problemas ZDT são geralmente testados com $N_E = 25000$ avaliações (ZHANG; LI, 2007). Além disso, para verificar o comportamento dos algoritmos ELMOEA/D em diferentes dimensões, também foram analisadas as aproximações geradas nos problemas ZDT com $N_D = 8$ variáveis e $N_E = 200$

avaliações; e com o dobro do número de variáveis sugerido ($N_D = 60$ para ZDT1, ZDT2 e ZDT3 e $N_D = 20$ para ZDT4 e ZDT6) e $N_E = 2000$ avaliações. Além dos dados anteriores, o ponto de referência (\mathbf{r}) adotado em cada problema é mostrado na Tabela 4.

Tabela 4: Parâmetros dos *benchmarks* e ELM

MOP	$N_E = 200$				$N_E = 1000$				$N_E = 2000$			
	N_D	\mathbf{r}	C	$h(\mathbf{x})$	N_D	\mathbf{r}	C	$h(\mathbf{x})$	N_D	\mathbf{r}	C	$h(\mathbf{x})$
ZDT1	8	(10, 10)	2^8	MQ	30	(5, 5)	2^4	MQ	60	(5, 5)	2^3	MQ
ZDT2	8	(10, 10)	2^9	SIG	30	(5, 5)	2^{-3}	MQ	60	(5, 5)	2^{-6}	MQ
ZDT3	8	(20, 20)	2^4	SIG	30	(5, 5)	2^5	MQ	60	(5, 5)	2^{-1}	MQ
ZDT4	8	(50, 50)	2^{-6}	MQ	10	(50, 50)	2^{-10}	MQ	20	(50, 50)	2^2	SIG
ZDT6	8	(10, 10)	2^6	SIG	10	(5, 5)	2^5	SIG	20	(5, 5)	2^3	MQ

O parâmetro regularizador C e a função de ativação da ELM $h(\cdot)$ são específicos para cada problema e foram encontrados pelo seguinte procedimento: com uma população gerada aleatoriamente com o LHS, do mesmo tamanho do conjunto de treinamento inicial (N_T), testam-se os valores $C = 2^{-10}, 2^{-9}, \dots, 2^{10}$ (baseados nos valores testados por Huang *et al.* (2012)) nas três funções de ativação citadas na Seção 2.5 (SIG, GAU e MQ). O erro de aproximação em cada configuração testada é calculado pela média dos erros em uma validação cruzada 5 – *fold*. A função de ativação é escolhida de acordo com o menor erro encontrado e o parâmetro regularizador $C = 2^k$ será o de menor erro no qual o próximo valor testado ($C = 2^{k+1}$) produz um erro maior, assim como na estratégia *early stopping* (HAYKIN, 2008). A função de ativação e valores do parâmetro C para cada problema são mostrados na Tabela 4.

Tabela 5: Parâmetros ELMOEA/D para os testes com *benchmarks*

MOEA/D		MOEA/D-DE	
G	100	N_R	2
N	300	Operador	DE/ <i>rand/1/bin</i>
N_B	20	CR	1
$F(\mathbf{x} \mathbf{w}, \mathbf{z})$	PBI	F	0,5
ELM		ELMOEA/D	
N_H	$2N_D + 1$	N_S	10 ($N_D = 8, 10, 30$) ou 20 ($N_D = 20, 60$)
N_T	$10N_D$	ϵ	10^{-2} ($N_D = 8$) ou 10^{-3} ($N_D = 10, 20, 30, 60$)

Os parâmetros do ELMOEA/D foram baseados no MOEA/D-RBF e na literatura relacionada. A Tabela 5 mostra os parâmetros do ELMOEA/D e seus valores para esta fase. Os parâmetros de cada algoritmo MOEA/D, como o tamanho da população ($N = 300$), número

de gerações ($G = 100$), número de vizinhos ($N_B = 20$), função de agregação PBI foram os mesmos utilizados no MOEA/D-RBF (MARTÍNEZ; COELLO, 2013). Assim como no algoritmo MOEA/D-RBF, o tamanho do arquivo de treinamento é definido como $N_T = 10N_D$ e o número de soluções selecionadas para avaliação em cada iteração é $N_S = 10$, exceto nos problemas com o dobro do número de variáveis onde $N_S = 20$. Apesar dos dois algoritmos utilizarem modelos *surrogate* diferentes (três redes RBF no MOEA/D-RBF e uma rede ELM no ELMOEA/D proposto), o mesmo número de neurônios ocultos foi especificado ($N_H = 2N_D + 1$). Os pesos da camada de entrada do modelo ELM são selecionados aleatoriamente no intervalo $[-1; 1]$. Além destes, o operador *DE/rand/1/bin* nas variantes MOEA/D-DE e MOEA/D-STM possui os valores clássicos para os parâmetros de escalarização ($F = 0,5$) e taxa de cruzamento ($CR = 1$) (LI; ZHANG, 2009).

As aproximações são avaliadas com a métrica Hipervolume (Equação 35), e número de avaliações (N_E) de cada problema são mostrados na Tabela 4. A Tabela 6 mostra a média e desvio-padrão para a métrica Hipervolume, bem como o tempo de execução para a plataforma ELMOEA/D, com diferentes variantes do algoritmo MOEA/D: MOEA/D original (Seção 2.3), MOEA/D-DE (Seção 2.3.1) e MOEA/D-STM (Seção 2.3.2). Em resumo, os resultados mostram que:

- nos MOPs com 8 variáveis, houve pouca diferença entre as três variantes. O MOEA/D-STM é melhor no problema ZDT2, e nenhuma variante atinge bons resultados nos MOPs ZDT4 e ZDT6, comparados aos resultados para $N_D = 10$ e $N_E = 1000$;
- nos MOPs com 30 e 10 variáveis, a variante do algoritmo ELMOEA/D com MOEA/D original possui os melhores resultados nos problemas ZDT3 e ZDT4 e empata com MOEA/D-DE no problema ZDT2. A variante com o algoritmo MOEA/D-DE está entre os melhores apenas nos problemas ZDT1 e ZDT2. A variante com MOEA/D-STM, por sua vez, não possui bom desempenho no problema ZDT2 e é a melhor para o MOP ZDT6, apesar de obter maior variância;
- nos MOPs com 60 e 20 variáveis, a variante com MOEA/D original é a melhor no problema ZDT2, possui os melhores resultados no problema ZDT3 e está entre os melhores no MOP ZDT6. O MOEA/D-STM possui clara vantagem no problema ZDT1 e, ainda que com maior variância, empata com MOEA/D original no MOP ZDT6. Nenhuma variante encontrou soluções dominando o ponto de referência no MOP ZDT4 com 20 variáveis.

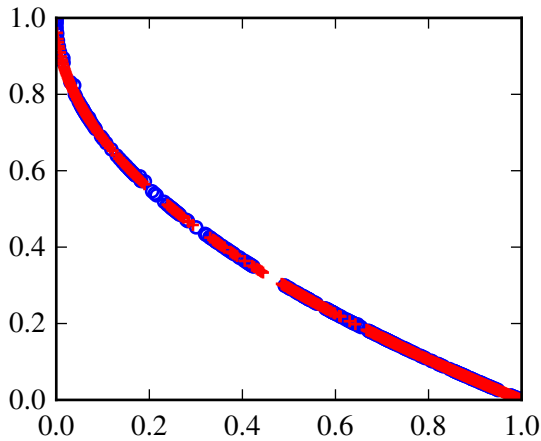
Com base nestes resultados, observa-se que:

Tabela 6: Comparação entre variantes do algoritmo MOEA/D na plataforma ELMOEA/D: média e desvio padrão para a métrica Hipervolume e tempo de execução (blocos sombreados indicam a melhor variante (em negrito) segundo o teste de Kruskal-Wallis).

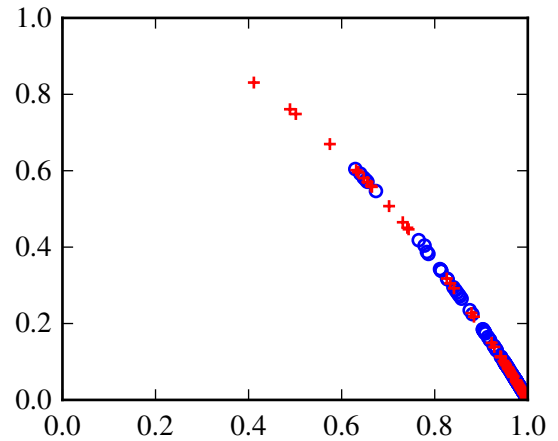
MOP	N_D	ELMOEA/D e MOEA/D		ELMOEA/D e MOEA/D-DE		ELMOEA/D e MOEA/D-STM	
		Hipervolume (I_H^-)	Tempo(s)	Hipervolume (I_H^-)	Tempo(s)	Hipervolume (I_H^-)	Tempo(s)
ZDT1	8	0,004601 ± 0,000416	1.32 ± 0.18	0,004515 ± 0,000512	1,49 ± 0,07	0,004487 ± 0,000456	66,7 ± 7,46
ZDT2	8	0,010523 ± 0,010363	1.57 ± 0.25	0,007058 ± 0,005156	1,68 ± 0,11	0,005338 ± 0,004168	68,6 ± 9,66
ZDT3	8	0,052846 ± 0,056233	1.52 ± 0.24	0,084763 ± 0,142886	1,67 ± 0,08	0,078723 ± 0,124463	69,0 ± 7,36
ZDT4	8	1706,449 ± 372,2544	1.04 ± 0.24	1568,255 ± 348,1619	1,58 ± 0,60	1564,853 ± 493,9632	99,5 ± 41,1
ZDT6	8	0,814625 ± 1,161673	1.58 ± 0.66	1,137031 ± 1,551299	2,39 ± 0,96	1,119733 ± 1,466206	119 ± 62,8
ZDT1	30	0,005104 ± 0,000949	21.6 ± 3.59	0,003535 ± 0,001243	14,0 ± 0,95	0,003309 ± 0,000942	352 ± 19,2
ZDT2	30	0,004206 ± 0,001808	20.9 ± 2.98	0,002799 ± 0,001153	13,0 ± 0,12	0,857418 ± 1,425233	376 ± 29,2
ZDT3	30	0,416961 ± 0,320079	24.6 ± 5.12	0,650522 ± 0,332326	15,5 ± 2,00	0,662240 ± 0,550968	368 ± 25,3
ZDT4	10	1073,470 ± 229,1957	12.8 ± 1.73	1171,162 ± 326,2744	11,1 ± 1,15	1214,550 ± 234,9603	472 ± 78,3
ZDT6	10	0,248175 ± 0,369773	8.99 ± 1.15	0,059536 ± 0,064651	11,5 ± 1,21	0,056692 ± 0,125754	426 ± 76,5
ZDT1	60	0,006974 ± 0,003693	59.2 ± 8.49	0,003481 ± 0,004247	30,2 ± 3,92	0,000566 ± 0,000480	483 ± 143
ZDT2	60	0,013782 ± 0,004571	55.1 ± 7.32	0,017142 ± 0,004158	32,7 ± 3,96	0,017689 ± 0,010067	499 ± 155
ZDT3	60	0,562457 ± 0,245252	60.6 ± 10.2	0,718939 ± 0,318612	61,9 ± 10,1	0,706681 ± 0,353594	497 ± 145
ZDT4	20	2499,666 ± 0,000000	14.3 ± 1.67	2499,666 ± 0,000000	24,4 ± 4,27	2499,666 ± 0,000000	589 ± 77,4
ZDT6	20	0,058311 ± 0,133820	13.1 ± 1.73	0,083678 ± 0,123224	14,5 ± 1,50	0,133948 ± 0,453151	606 ± 739

- a variante com MOEA/D-STM gera melhor cobertura no problema ZDT1. Como os valores pequenos da métrica Hipervolume indicam, a convergência no problema ZDT1 é simples, mesmo em poucas gerações. Aparentemente, a variante MOEA/D-STM ainda consegue gerar várias soluções não-dominadas mesmo após vários pontos não-dominados serem encontrados, cobrindo a maior parte da \mathcal{PF} , como mostrado no exemplo de frente encontrada pelos algoritmos na Figura 13a e pela função de *attainment* na Figura 14.
- além disso, o algoritmo MOEA/D-STM produz bons resultados no problema ZDT6 que possui distribuição de pontos não-uniforme. No problema ZDT6, várias soluções estão próximas ao ponto extremo da \mathcal{PF} (0; 1) e poucos pontos próximos ao outro extremo (0, 28; 0, 92). Como visto na Figura 13b e na função de *attainment* da Figura 15, mesmo na sua pior aproximação da \mathcal{PF} , a variante com o algoritmo MOEA/D-STM mantém boa diversidade em relação ao algoritmo MOEA/D no problema ZDT6 com 30 variáveis;
- o algoritmo MOEA/D original possui bons resultados no problema ZDT2 em relação às outras duas variantes (MOEA/D-DE e MOEA/D-STM). Uma explicação para este resultado pode ser o uso do operador genético *Simulated Binary Crossover*, que utiliza mais informações da vizinhança na escolha dos pais com o torneio binário e os novos indivíduos serão sempre próximos aos pais. Sindhya *et al.* (2011) já observa que a busca mais localizada (*exploitation*) melhora o desempenho no problema ZDT2. O mesmo vale para o problema ZDT3, onde é possível que os operadores da variante MOEA/D original estejam explorando melhor cada segmento da \mathcal{PF} ;
- a variante com MOEA/D original também se destaca no problema ZDT4 com 10 variáveis, caracterizado por conter muitas frentes de Pareto locais. Devido a esta característica, a função de agregação pode tornar-se não-linear e dificulta o progresso da busca. É possível que a atualização da vizinhança no MOEA/D original ajude a descartar mais rapidamente os indivíduos presos em frentes sub-ótimas em favor do novo indivíduo. A atualização da vizinhança no MOEA/D original substitui qualquer indivíduo da vizinhança desde que o valor da função de agregação seja menor em relação ao novo indivíduo, ao contrário da atualização da vizinhança mais restrita dos algoritmos MOEA/D-DE e MOEA/D-STM. Mais experimentos são necessários para suportar esta hipótese.

Em geral, a variante do ELMOEA/D com MOEA/D-STM apresenta os melhores resultados para a métrica Hipervolume no teste de Kruskal-Wallis. Entretanto, o tempo de execução é até 50 vezes maior comparado às variantes com MOEA/D original e MOEA/D-DE. Com isso, a variante ELMOEA/D com algoritmo MOEA/D original será considerada a melhor para os problemas ZDT.



(a) Exemplo de aproximação de frente ZDT1



(b) Exemplo de aproximação de frente ZDT6

Figura 13: Exemplo de \mathcal{PF} encontradas pelas variantes do algoritmo ELMOEA/D utilizando MOEA/D original (o) e MOEA/D-STM (+): (a) aproximações da \mathcal{PF} no problema ZDT1 com 30 variáveis que obtiveram valor da mediana para métrica Hipervolume e (b) aproximações da \mathcal{PF} no problema ZDT6 com 10 variáveis que obtiveram o pior valor para a métrica Hipervolume.

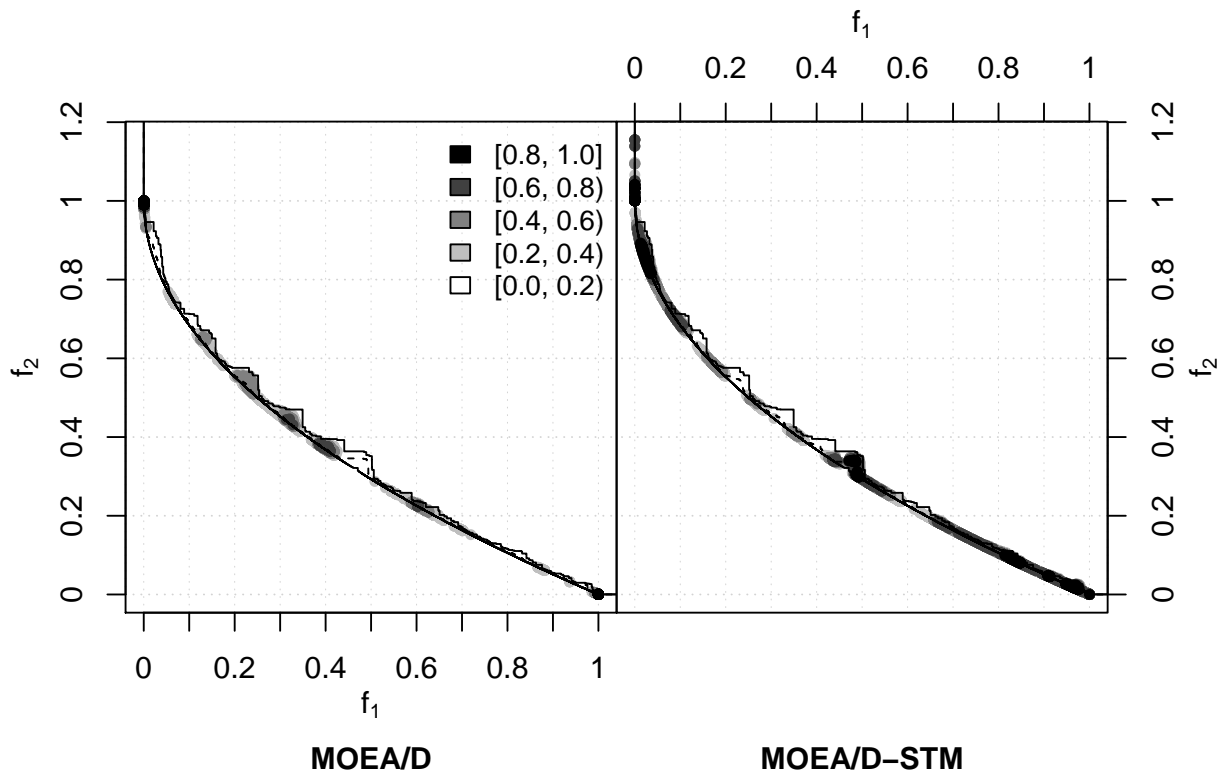


Figura 14: Função de *attainment* para o MOP ZDT1. O gráficos da esquerda e direita correspondem ao algoritmo MOEA/D e MOEA/D-STM. Regiões mais escuras representam maior chances de encontrar soluções para o dado algoritmo

Com o objetivo de analisar a vantagem do uso de *surrogates*, e mais especificamente *surrogates* baseados em ELMs, a Tabela 7 apresenta a média e desvio-padrão para a métrica

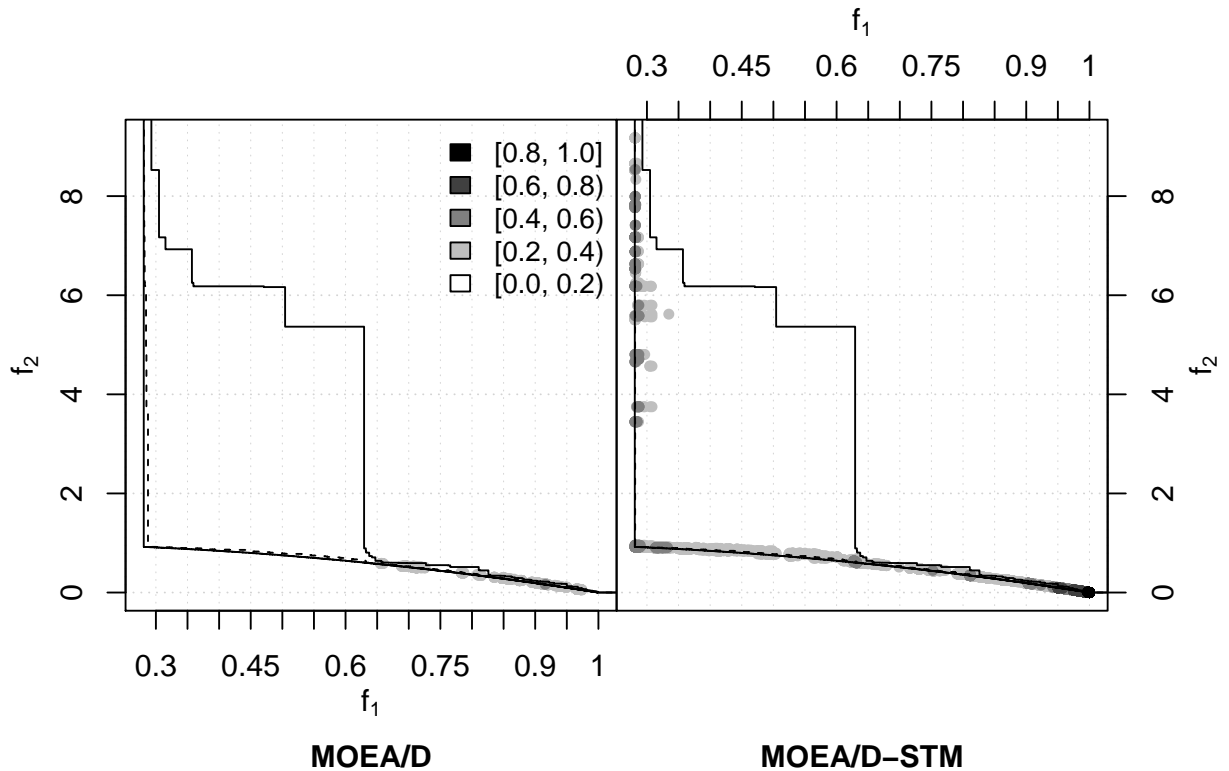


Figura 15: Função de *attainment* para o MOP ZDT6. Os gráficos da esquerda e direita correspondem ao algoritmo MOEA/D e MOEA/D-STM. Regiões mais escuras representam maior chance de encontrar soluções para o dado algoritmo

Hipervolume no algoritmo ELMOEA/D, MOEA/D-RBF e MOEA/D-DE sem *surrogate*. As mesmas configurações do ELMOEA/D com o algoritmo MOEA/D dos resultados anteriores foram utilizadas. O algoritmo que inspirou o ELMOEA/D, MOEA/D-RBF, utiliza o MOEA/D original descrito na Seção 2.3 e redes RBF como *surrogate*. Como base de comparação, os resultados do algoritmo MOEA/D-DE sem auxílio de *surrogates* também são apresentados. O algoritmo MOEA/D-DE sem *surrogates* utiliza uma população pequena $N = 100$, $N_B = 10$ e $N_R = 1$ para compensar o fato do número máximo de avaliações (N_E) e consequente número de gerações (G) ser reduzido.

Nota-se que os algoritmos ELMOEA/D e MOEA/D-RBF foram melhores que o algoritmo MOEA/D-DE sem *surrogates*. Além disso, o algoritmo ELMOEA/D se mostrou competitivo em relação ao MOEA/D-RBF nos problemas testados, com média estatisticamente melhor na maioria dos problemas e valores da métrica Hipervolume parecidos nos problemas ZDT2 e ZDT4 com 8 variáveis e ZDT1 com 30 variáveis.

Quanto à escalabilidade do algoritmo ELMOEA/D em relação ao número de variáveis, a proposta manteve bom desempenho nos MOPs com 60 e 20 variáveis na maioria dos problemas. Com exceção do problema ZDT2, que obteve média e variância do Hipervolume maiores para

Tabela 7: Comparação do ELMOEA/D com algoritmos da literatura: de acordo com a média e desvio padrão da métrica Hipervolume (fonte em negrito indica casos em que a média foi relativamente melhor).

MOP	N_D	ELMOEA/D	MOEA/D-RBF ^a	MOEA/D-DE ^b
ZDT1	8	0,004601 ± 0,000416	0,009170 ± 0,000890	14,60113 ± 3,101870
ZDT2	8	0,010523 ± 0,010363	0,009703 ± 0,001520	25,23048 ± 3,813456
ZDT3	8	0,052846 ± 0,056233	0,115358 ± 0,898476	38,24666 ± 5,888609
ZDT4	8	1706,449 ± 372,2544	1681,549 ± 469,3747	2305,849 ± 264,0081
ZDT6	8	0,814625 ± 1,161673	1,690381 ± 2,006687	66,46608 ± 5,244615
ZDT1	30	0,005104 ± 0,000949	0,004831 ± 0,000382	11,34819 ± 0,864443
ZDT2	30	0,004206 ± 0,001808	0,006654 ± 0,000744	18,48721 ± 1,247111
ZDT3	30	0,416961 ± 0,320079	2,298589 ± 0,952347	12,14161 ± 1,248740
ZDT4	10	1073,470 ± 229,1957	1828,494 ± 382,6049	2268,330 ± 272,4258
ZDT6	10	0,248175 ± 0,369773	1,244183 ± 0,996797	23,20209 ± 0,000000
ZDT1	60	0,006974 ± 0,003693	–	12,06770 ± 0,656699
ZDT2	60	0,013782 ± 0,004571	–	19,89481 ± 0,784954
ZDT3	60	0,562457 ± 0,245252	–	13,32311 ± 0,713744
ZDT4	20	2499,666 ± 0,000000	–	2499,666 ± 0,000000
ZDT6	20	0,058311 ± 0,133820	–	23,20209 ± 0,000000

^a Valores retirados diretamente de (MARTÍNEZ; COELLO, 2013).

^b MOEA/D-DE com as mesmas configurações e número de avaliações, $N = 100$, $N_B = 10$ e $N_R = 1$.

60 variáveis, comparado aos resultados para o mesmo problema com 30 variáveis, e do problema ZDT4, em que nenhum ponto da aproximação domina o ponto de referência $\mathbf{r} = (50; 50)$.

Portanto, com os experimentos realizados, pode-se concluir que o uso de *surrogates* é importante quando se tem poucos recursos para avaliação do *fitness*, e mais, o uso de modelos ELM como *surrogates* é promissor e, comparado ao uso de *ensemble* de redes RBF, obtém melhores resultados com modelos mais simples. A proposta também foi capaz de manter o desempenho satisfatório com o dobro do número de variáveis. Os resultados apresentados nesta seção corroboram as conclusões apresentadas em um trabalho com resultados preliminares (PAVELSKI *et al.*, 2014).

4.3 PARTE II: PPEP MULTIOBJETIVO

Como indicado na Seção 2.6, o problema PPEP pode ser visto como um MOP biobjetivo para o qual se busca minimizar a energia dos termos ligados e não-ligados do sistema. A formulação do PPEP como MOP possui vantagens como simular interações entre diferentes moléculas durante a otimização. Nessa seção, os resultados alcançados pelo algoritmo ELMOEA/D para o PPEP são apresentados e discutidos.

Os testes consistem em prever a estrutura de sete proteínas conhecidas. Os identificadores PDB e algumas características de cada proteína são apresentadas na Tabela 8. As mesmas proteínas são utilizadas por Cutello *et al.* (2006), e possuem tamanhos de 5 a 74 resíduos, resultando em 17 a 275 variáveis (N_D).

Tabela 8: Características das proteínas utilizadas nos testes com PPEP multiobjetivo.

Proteína	Id. PDB	N. de resíduos	N. de átomos	N_D	Tempo médio de avaliação (ms)
<i>Met-Enkephalin</i>	1PLW	5	75	17	$19 \pm 1,7$
<i>Melittin</i>	2MLT	26	402	99	$28 \pm 1,8$
<i>Disulphide-stabilized mini protein A domain</i>	1ZDD	34	566	146	$34 \pm 1,8$
<i>Crambin</i>	1CRN	46	327	145	$36 \pm 1,8$
<i>Repressor of primer</i>	1ROP	63	447	253	$52 \pm 2,0$
<i>Uteroglobin</i>	1UTG	70	548	275	$58 \pm 2,1$
<i>Ribosomal Protein L7/L12</i>	1CTF	74	487	271	$55 \pm 1,9$

O número de variáveis e o tempo médio relativo a uma avaliação de cada proteína também é apresentado na Tabela 8. Apesar do tempo de uma avaliação não parecer proibitivo, problemas deste tipo podem exigir um número de avaliações na ordem de 10^5 avaliações (CUTELLO *et al.*, 2006). Deste modo, problemas como a predição de estruturas para proteínas grandes e aplicações que exigem o *design* de proteínas para dada função (TRAMONTANO, 2006) podem se beneficiar da otimização de estruturas com o menor tempo possível. Em todos os testes com PPEP, o número máximo de avaliações da função-objetivo é restringido a $200N_D$. O tempo computacional total para geração dos resultados dessa seção foi aproximadamente 15 dias.

4.3.1 MELHOR VARIANTE E COMPARAÇÃO COM MOEA/D-DE SEM *SURROGATE*

Primeiramente, assim como feito para os MOPs *benchmark*, a proposta foi testada com as variantes do MOEA/D, neste caso, MOEA/D original e MOEA/D-DE. Como o algoritmo MOEA/D-STM apresentou custo computacional relativamente elevado, a variante não foi considerada neste teste. As comparações são feitas com três proteínas pequenas (peptídeos 1PLW, 2MLT e 1ZDD). Após determinar a melhor variante, a proposta é comparada com o algoritmo sem *surrogates*. Nos dois casos, as populações finais dos algoritmos serão avaliadas segundo a métrica $DM(\cdot)$ (Equação 36) e os resultados são analisados por meio dos gráficos *boxplot* e teste de Kruskal-Wallis.

A Figura 16 apresenta os resultados da comparação entre as duas variantes do MOEA/D.

Na proteína 1PLW, as variantes empatam, porém nas proteínas 2MLT e 1ZDD a variante com MOEA/D-DE foi superior. Como apontado por Li e Zhang (2009), os operadores DE são mais apropriados para problemas com \mathcal{PS} complexos, como o conjunto de problemas UF1-10. De fato, as proteínas com estruturas mais complexas (2MLT e 1ZDD) que possivelmente possuem \mathcal{PS} mais complexos, obtiveram os melhores resultados com a estratégia MOEA/D-DE.

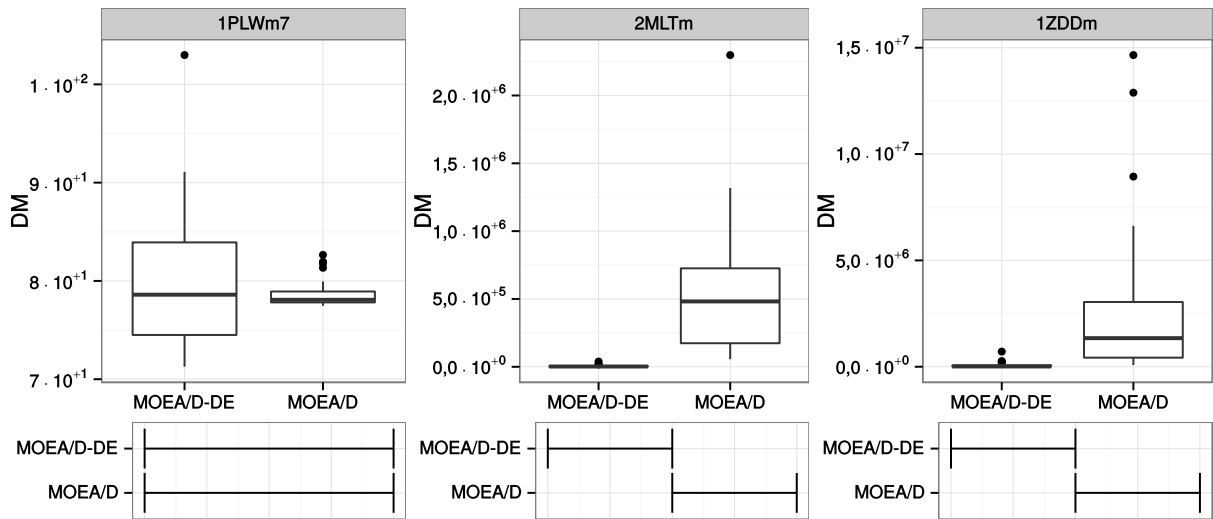


Figura 16: Resultados para diferentes variantes do otimizador MOEA/D no algoritmo EL-MOEA/D: comparação entre as variantes com o algoritmo MOEA/D original e MOEA/D-DE.

A Figura 17 mostra os resultados da comparação entre o algoritmo proposto EL-MOEA/D e o algoritmo MOEA/D-DE sem *surrogates*. Para os dois algoritmos as mesmas configurações utilizadas nos problemas *benchmarks* foram adotadas para o PPEP, exceto para os parâmetros C e $h(\mathbf{x})$ da ELM que foram reajustados. A melhor configuração para ELM foi descoberta empiricamente, sendo a função de ativação gaussiana (GAU) com um valor pequeno para o parâmetro regularizador $C = 2^{-10}$ e as entradas para o treinamento normalizadas para o intervalo $[-1; 1]$.

Observa-se que, nos parâmetros dados, o algoritmo MOEA/D-DE sem *surrogates* supera o ELMOEA/D em todos os casos, ou seja, em dadas condições o uso de *surrogates* não proporciona melhores resultados. A readequação dos parâmetros será feita com as três proteínas menores 1PLW, 2MLT e 1ZDD e, após o ajuste (Seção 4.3.2), a proposta com os melhores parâmetros será novamente comparada ao algoritmo MOEA/D-DE sem *surrogates* (Seção 4.3.3).

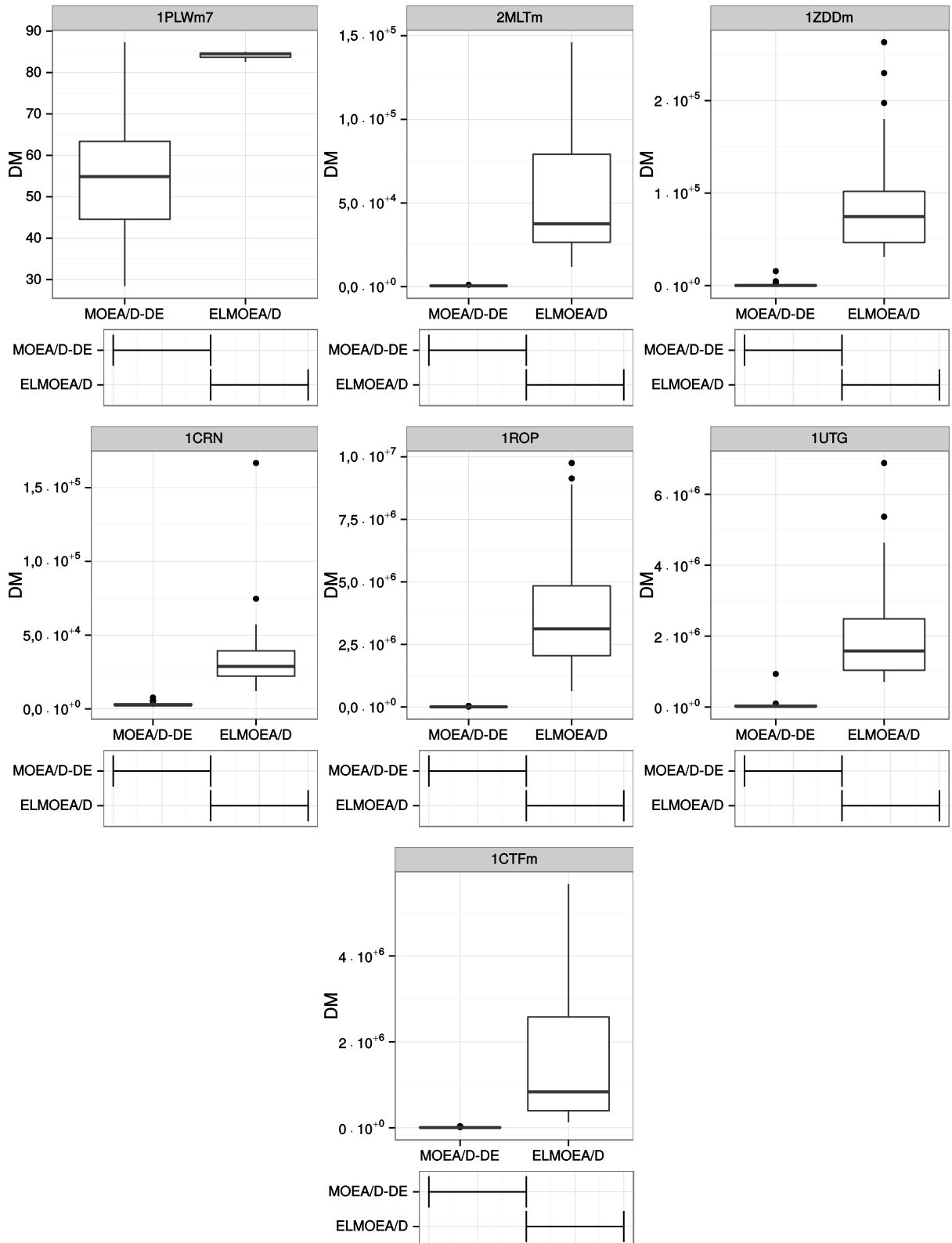


Figura 17: Comparação do algoritmo ELMOEA/D e MOEA/D nas configurações iniciais, utilizadas nos problemas *benchmarks*.

4.3.2 AJUSTE DE PARÂMETROS

Devido ao fato da resolução do problema PPEP ser inédita para algoritmos MOEA/D assistidos por *surrogates* baseados em ELM, os resultados dos testes para escolha de parâmetros podem ser interessantes em trabalhos futuros. Os testes de diferentes parâmetros para a plataforma ELMOEA/D também são conduzidos para problemas benchmark ZDT com o objetivo de auxiliar a análise para o PPEP. Isto porque os problemas ZDT, diferente do PPEP, possuem \mathcal{PF} conhecidas e propriedades amplamente investigadas pela comunidade de pesquisa em MOEAs. Assim, a análise dos resultados do ajuste de parâmetros para os benchmarks pretende auxiliar no entendimento dos resultados obtidos para o PPEP. As tabelas com estes resultados estão incluídas no Apêndice A.

Em cada comparação, os parâmetros testados são descritos e as qualidades das aproximações são analisadas com auxílio de gráficos *boxplot*, de acordo com os valores da métrica $DM(\cdot)$ em cada aproximação (eixo y dos gráficos), e teste de Kruskal-Wallis. A Tabela 9 resume os parâmetros e valores analisados, bem como o valor que obtém melhores resultados.

Tabela 9: Resultados dos testes com diferentes parâmetros e configurações para o PPEP das Seções 4.3.2 e 4.3.4.

Parâmetro (por algoritmo)	Valores testados	Fig.	Melhor
1) Tamanho da população (N) e número de gerações (G)	100 e 1, 300 e 1, 100 e 10, 300 e 10	18	100 e 1
2) Função de agregação ($F(\cdot)$)	PBI, TCHE, NTCHE	19	NTCHE
3) Operador DE e Seleção adaptativa	DE/ <i>rand/1/bin</i> , DE/ <i>rand/2/bin</i> , DE/ <i>nonlinear</i> , PM, FFRMAB	20	DE/ <i>nonlinear</i>
4) Número de <i>surrogates</i> (N_a)	1, 2, 5, 10	21	1
5) Número de objetivos (N_O)	2, 7	24	1

Primeiramente, a influência dos parâmetros do MOEA/D-DE é investigada: o número de gerações (G) e o tamanho da população (N), bem como os consequentes tamanho de vizinhança (N_B) e número de substituições (N_R) (visto que, em geral, utiliza-se $N_B = 0.1N$ e $N_R = 0.01N$ (LI *et al.*, 2014b)). Todas as combinações de uma única geração ($G = 1$) ou mais gerações ($G = 10$) com uma população pequena ($N = 100$, $N_B = 10$ e $N_R = 1$) e grande ($N = 300$, $N_B = 20$ e $N_R = 2$) são testadas. A Figura 18 apresenta o *boxplot* e resultados para o teste Kruskal-Wallis nas três proteínas. A configuração com única geração e população pequena (1 – 100) mostra os melhores resultados: empata com a configuração de única geração e população grande (1 – 300) nas proteínas 1PLW e 1ZDD, mas supera na proteína 2MLT.

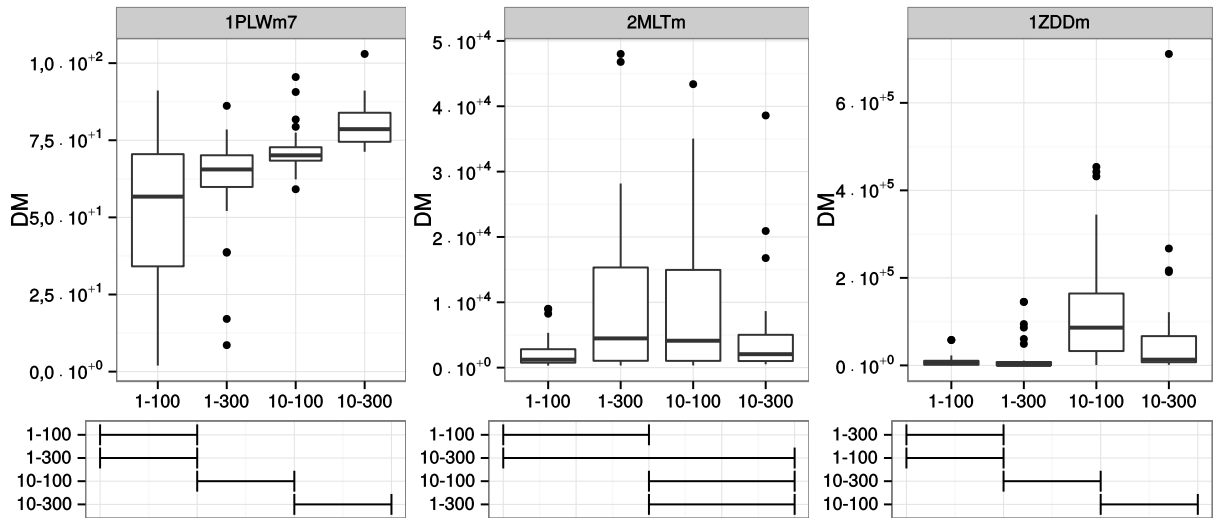


Figura 18: Resultados para diferentes tamanhos de população e número de gerações do otimizador MOEA/D-DE no algoritmo ELMOEA/D: comparação entre única geração com população pequena (1 – 100) ou grande (1 – 300) ou 10 gerações com população pequena (10 – 100) ou grande (10 – 300).

Nos problemas *benchmark* (Tabela 10, Apêndice A), a maioria dos problemas se beneficia de uma população grande ($N = 300$) e muitas gerações ($G = 100$) com evolução baseada no cálculo aproximado dos objetivos (*surrogate*). Porém, é importante observar que no problema ZDT4, que possui múltiplas \mathcal{PF} locais, a população menor e poucas gerações com a aproximação dos *surrogates* possui melhores resultados. Com isso, há indícios de que populações grandes e muitas gerações para o MOEA/D levam a convergência prematura em ótimos locais. Enquanto que, populações pequenas com poucas gerações produzem melhores resultados em MOPs com várias frentes locais sub-ótimas, no qual o PPEP pode estar incluído.

A função de agregação possui influência não apenas no otimizador MOEA/D-DE, mas também na seleção das soluções candidatas e atualização do arquivo de treinamento do algoritmo ELMOEA/D. As três abordagens mencionadas são comparadas, TCHE, NTCHE e PBI, mantendo a melhor configuração anterior ($G = 1$, $N = 100$). Como mostra a Figura 19, a função de agregação NTCHE obtém as menores medianas nas três proteínas e é significativamente melhor que TCHE e PBI na proteína 1PLW.

Quanto aos MOPs *benchmark* (Tabela 11 do Apêndice A), a função de agregação PBI produz as melhores aproximações. O sucesso relativo da estratégia NTCHE nos problemas ZDT3 e ZDT4 com $N_D = 30$ e $N_D = 10$ pode ser devido ao fato desses problemas apresentarem maiores disparidades dos intervalos de valores de cada objetivo. Na população inicial para estes problemas, o primeiro objetivo está no intervalo $[0; 1]$ e o segundo objetivo está, para o problema

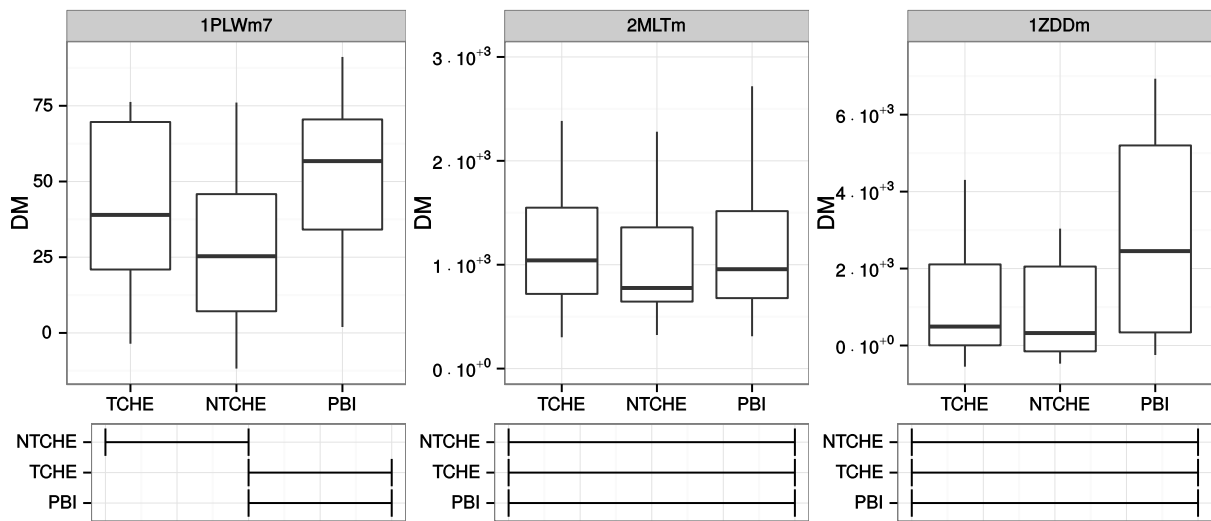


Figura 19: Resultados para diferentes funções de agregação no algoritmo ELMOEA/D: comparação entre as abordagens TCHE, NTCHE e PBI.

ZDT3, dentro de $[1,5;6,5]$ e, para o MOP ZDT4, entre $[80;240]$. O PPEP também possui intervalos de objetivos díspares e se beneficia da função de agregação normalizada NTCHE.

Os operadores DE influenciam o modo como o otimizador MOEA/D-DE explora o espaço de soluções das funções-objetivo estimadas para gerar soluções candidatas a avaliação com a função custosa. Os operadores *DE/rand/1/bin*, *DE/rand/2/bin* e *DE/nonlinear* são comparados, bem como as estratégias de seleção adaptativa de operadores PM e FFRMAB. As estratégias PM e FFRMAB adaptam a decisão de selecionar um dos três operadores DE (*DE/rand/1/bin*, *DE/rand/2/bin* e *DE/nonlinear*) para gerar o novo indivíduo (Seção 2.4) e seus parâmetros foram determinados com testes prévios. Na estratégia PM, a recompensa é normalizada com os valores extremos (ExtNorm, como visto em (GONG *et al.*, 2011)), $p_{min} = 0.1$ e $\alpha = 0.8$; na estratégia FFRMAB, $N_{sw} = 0.5N$, $C_s = 1.0$ e $D = 1.0$. Nas duas estratégias de seleção adaptativa, os valores de recompensa e probabilidades são inicializados no começo do algoritmo e mantidos entre as iterações do ELMOEA/D.

Os resultados da Figura 20 mostram que o operador *DE/nonlinear* é o melhor na proteína 2MLT e está entre os melhores na instância 1ZDD. Nas condições testadas, as estratégias de seleção adaptativa de operadores não são superiores àquelas que usam o operador *DE/nonlinear* de forma fixa.

De acordo com os resultados na Tabela 12 do Apêndice A, quando aplicados para resolver os problemas ZDT, o melhor operador dentre todos é o *DE/rand/1/bin*. A seleção adaptativa de operadores FFRMAB atinge desempenho semelhante ao operador *DE/rand/1/bin*

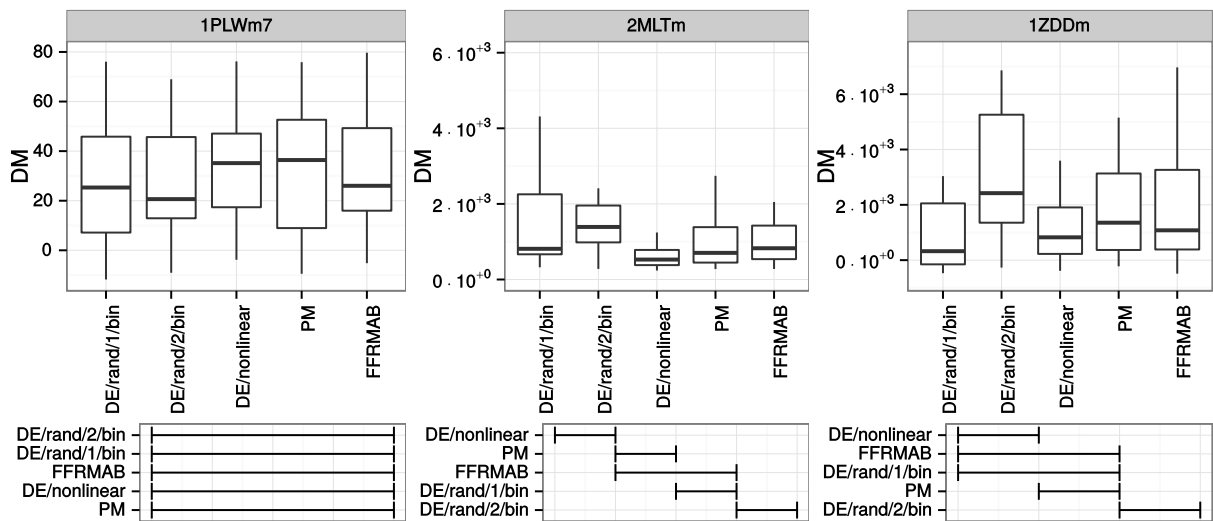


Figura 20: Resultados para diferentes operadores DE e hiper-heurísticas no algoritmo ELMOEA/D: comparação entre os operadores *DE/rand/1/bin*, *DE/rand/2/bin*, *DE/nonlinear* e hiper-heurísticas PM e FFRMAB.

na maioria dos casos. Os resultados também refletem o que é demonstrado por Sindhya *et al.* (2011), que a busca mais localizada do operador *DE/nonlinear* pode beneficiar os problemas ZDT2 e ZDT3. Não fica claro porque o modo de geração de novos indivíduos do operador *DE/nonlinear* gera melhores resultados no PPEP. Porém, é importante notar que este operador foi proposto para melhorar o desempenho de MOEAs quando há interdependências não-lineares entre indivíduos, podendo ser este o caso do PPEP.

Comparado aos resultados para os *benchmarks*, as estratégias de seleção adaptativa de operadores não apresentaram o mesmo comportamento no PPEP. Acredita-se que em uma geração ($G = 1$) não há muitas oportunidades da estratégia realizar a adaptação. Além disso, como em cada iteração do algoritmo ELMOEA/D, uma nova função f_a é gerada para aproximar os objetivos, as diferentes aproximações podem introduzir erro na adaptação através dos valores de recompensas.

A utilização de múltiplos modelos *surrogates*, como proposta na Seção 3.2.6, divide o espaço de objetivos de acordo com os pesos designados a cada *surrogate*. O desempenho da proposta é analisado com $N_a = 2, 5$ e 10 *surrogates* e $N_{a,T} = \lfloor 1, 5N_T/N_a \rfloor$ soluções para treinar cada ELM. De acordo com a Figura 21, apenas na proteína 2MLT há diferença significativa e a utilização de 1 ou 2 *surrogates* é melhor que 5 ou 10. Pode-se afirmar que 1 *surrogate* obtém a pior mediana nos resultados da proteína 1PLW, porém a diferença é pequena para 2 *surrogates*, e na proteína 1ZDD, 1 *surrogate* ainda possui a menor variância. Portanto, a utilização de 1 *surrogate* é mais adequada.

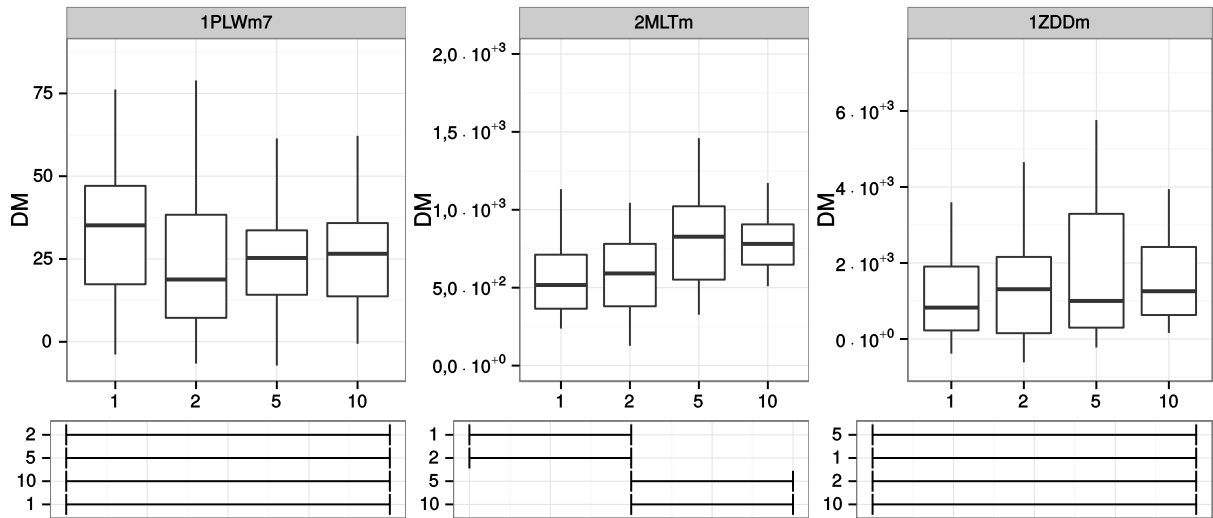


Figura 21: Resultados para múltiplas ELMs como *surrogates* algoritmo ELMOEA/D: comparação entre 1, 2, 5 e 10 *surrogates*.

Os resultados da Tabela 13 do Apêndice A também mostram que a utilização de múltiplos modelos *surrogate* não melhora os resultados na maioria dos problemas *benchmark*. Entretanto, os resultados para 2, 5 e 10 *surrogates* são estatisticamente melhores para o problema com \mathcal{PF} descontínua ZDT3 e no problema com múltiplas frentes sub-ótimas ZDT4. Seria razoável acreditar que, no problema ZDT3, cada *surrogate* ao final da busca estaria especializado em um segmento da \mathcal{PF} do problema ZDT3 (Figura 12c). Para o problema ZDT4, cada modelo *surrogate* pode estar adaptado para uma sub-frente no decorrer da busca.

4.3.3 COMPARAÇÃO FINAL COM MOEA/D-DE SEM *SURROGATES*

Finalmente, a melhor configuração da proposta ELMOEA/D para o PPEP é comparada ao algoritmo MOEA/D-DE sem *surrogates* e o resultado para todas as proteínas é apresentado na Figura 22. Observa-se que o ELMOEA/D obteve resultados melhores que o MOEA/D-DE nas proteínas 1PLW e 1CRN. Ainda, os dois algoritmos empatam, segundo o teste de Kruskal-Wallis, nas instâncias do PPEP com as proteínas 1ROP e 1UTG, sendo que ELMOEA/D possui as melhores medianas. No entanto, o MOEA/D-DE sem *surrogates* ainda supera a proposta nas proteínas 2MLT, 1ZDD e 1CTF.

Em relação ao desempenho ao longo do tempo, a Figura 23 apresenta a comparação dos algoritmos ELMOEA/D e MOEA/D-DE sem *surrogate*. Os gráficos representam uma curva suavizada, com a média e desvio padrão dos valores de qualidade das soluções (métrica $DM(\cdot)$) dos algoritmos ELMOEA/D (linha azul) e MOEA/D-DE (linha vermelha), em cada instante de

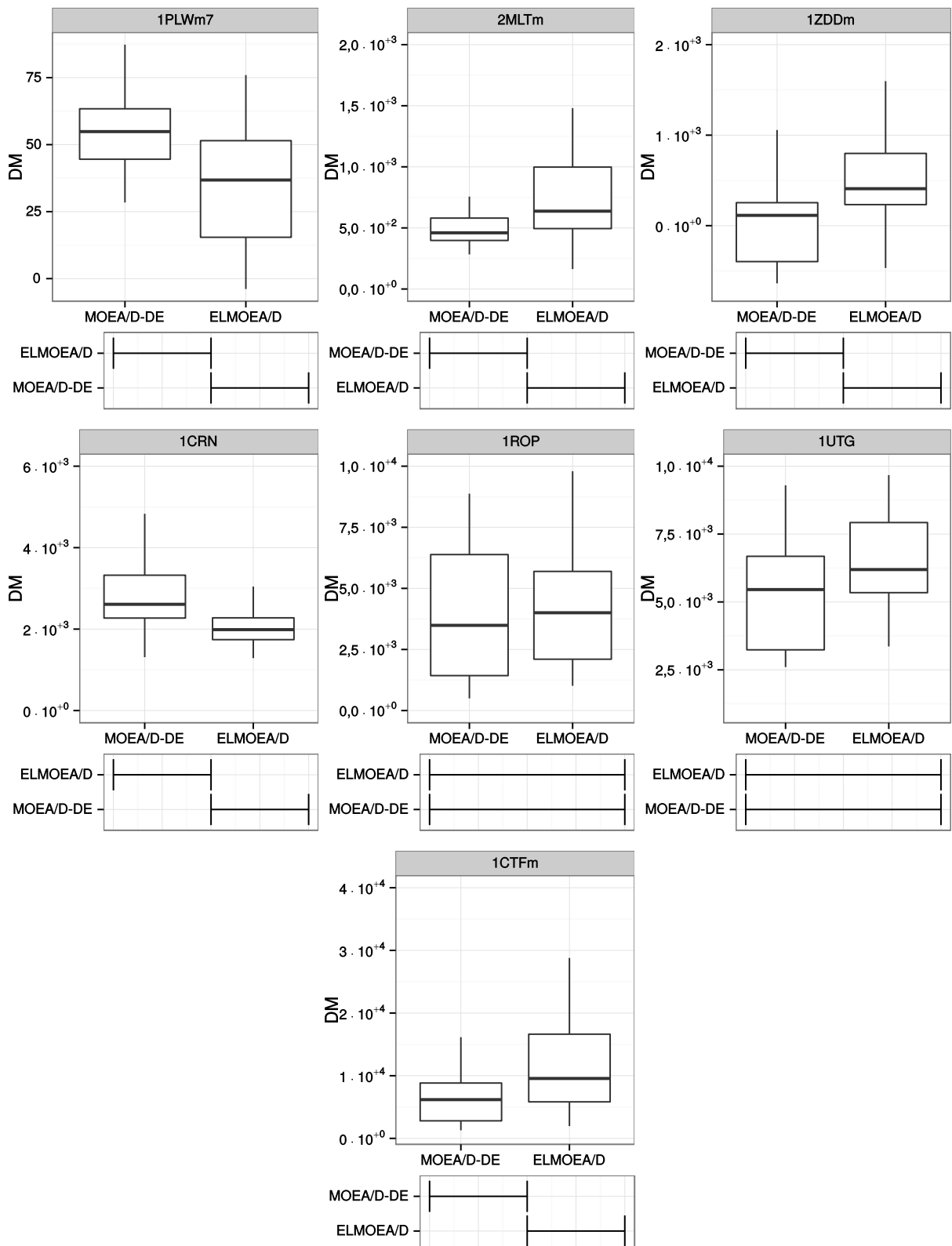


Figura 22: Comparação do algoritmo ELMOEA/D e MOEA/D-DE nas configurações finais, após o ajuste de parâmetros.

tempo (em segundos) nas 30 execuções. Nota-se que a proposta ELMOEA/D foi melhor que o algoritmo MOEA/D-DE em todos os instantes nas proteínas 1PLW e 1CRN. Nas proteínas 2MLT, 1ZDD e 1UTG, o algoritmo ELMOEA/D foi melhor que o MOEA/D-DE até 1/3 da execução. Os piores casos para o ELMOEA/D foram nas proteínas 1ROP e 1CTF, onde o algoritmo MOEA/D-DE produz melhores resultados durante todo o intervalo de tempo.

A utilização de *surrogates*, ao contrário do que foi observado nos problemas *benchmark*, não melhora os resultados em todos os casos no PPEP. No entanto, as observações feitas na aplicação da estratégia nos problemas *benchmarks* e ajuste de parâmetros apontam vários aspectos da proposta que poderiam ser melhorados para tornar o algoritmo mais robusto em diferentes cenários, como:

- adaptar o número de gerações do MOEA/D automaticamente, de acordo com o custo da função real;
- fazer com que objetivos com diferentes escalas e escalas semelhantes possuam boa performance, independente da função de agregação;
- melhorar a seleção automática de operadores, de modo que considere o erro introduzido pela função aproximada pelos *surrogates*;
- fazer com que a estratégia com múltiplos *surrogates* não perca desempenho em problemas com frentes de Pareto contínuas.

4.3.4 PPEP COM SETE OBJETIVOS

Como teste final, o comportamento do algoritmo ELMOEA/D no problema PPEP com 7 objetivos é observado. A formulação da Equação 25 é modificada para que cada energia constitua um objetivo. A expectativa do teste é verificar se o aprendizado dos *surrogates* melhora, pois cada energia individual será aproximada por uma saída j (associada a um conjunto de pesos de saída β_j) da rede ELM.

Para manter o mesmo tamanho dos conjuntos de vetores de pesos utilizados nos testes anteriores ($N = 100, N_T = 10N_D, N_S = 10$), os vetores de pesos são selecionados aleatoriamente no plano normalizado (a soma dos pesos é igual 1). A mesma estratégia é sugerida por (YUAN *et al.*, 2014) na geração de pesos para MOPs com muitos objetivos ($N_O > 4$).

A Figura 24 mostra que a utilização de sete objetivos não é melhor em nenhum dos casos. São necessários mais testes por parte do algoritmo MOEA/D-DE que não é especializado em problemas com muitos objetivos. Além disso, a disparidade entre os valores das energias

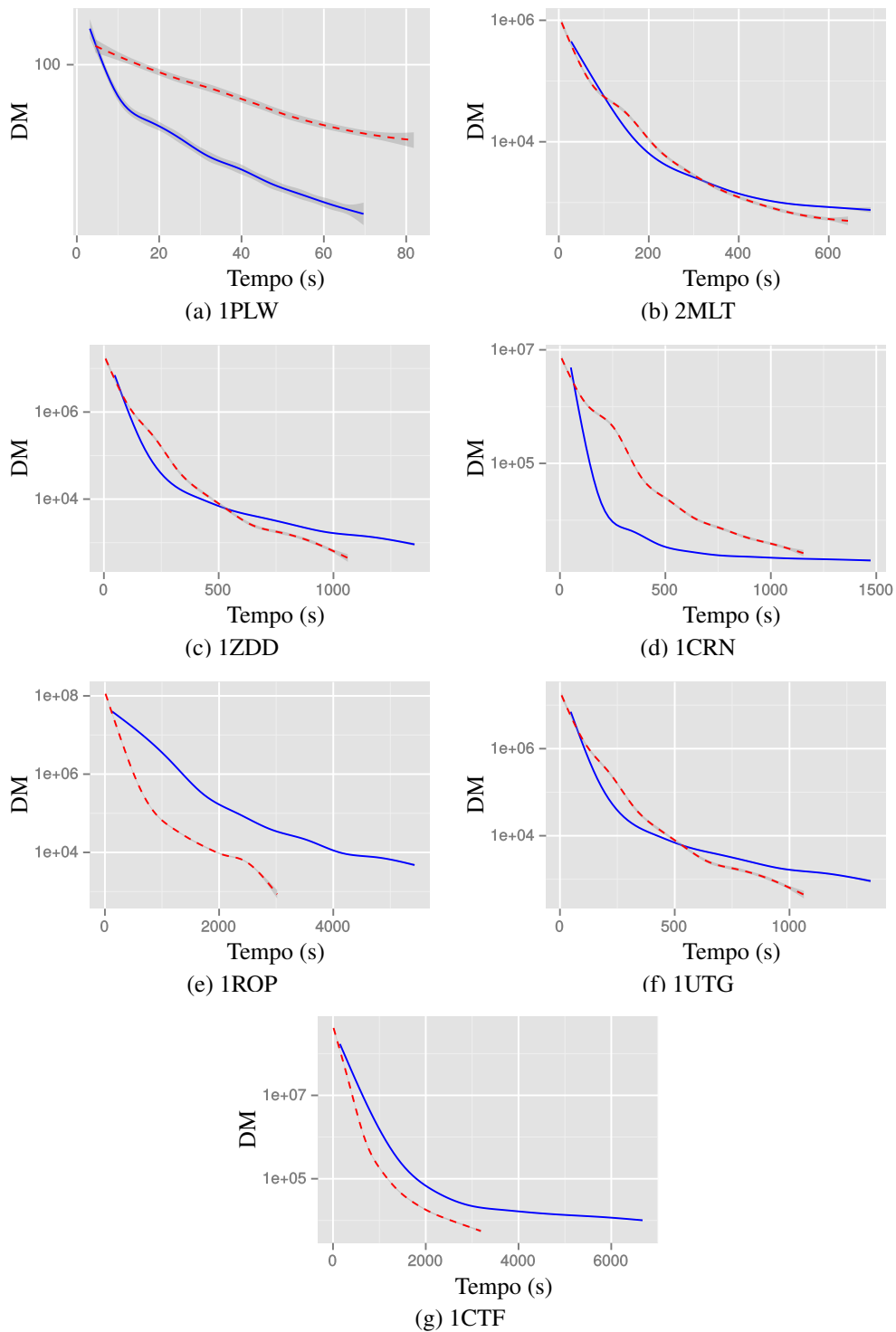


Figura 23: Comparação dos algoritmos ELMOEA/D e MOEA/D ao longo do tempo: curvas suavizadas adaptadas para os valores de qualidade $DM(\cdot)$ dos algoritmos ELMOEA/D (linha azul) e MOEA/D-DE (linha vermelha tracejada) em cada instante de tempo (segundos).

afeta a função de agregação que, nesse caso, a energia eletrostática predomina sobre as demais energias durante a substituição de soluções na população. Mais atenção também é necessária na reinicialização da população inicial, que depende da dominância de Pareto, visto que a

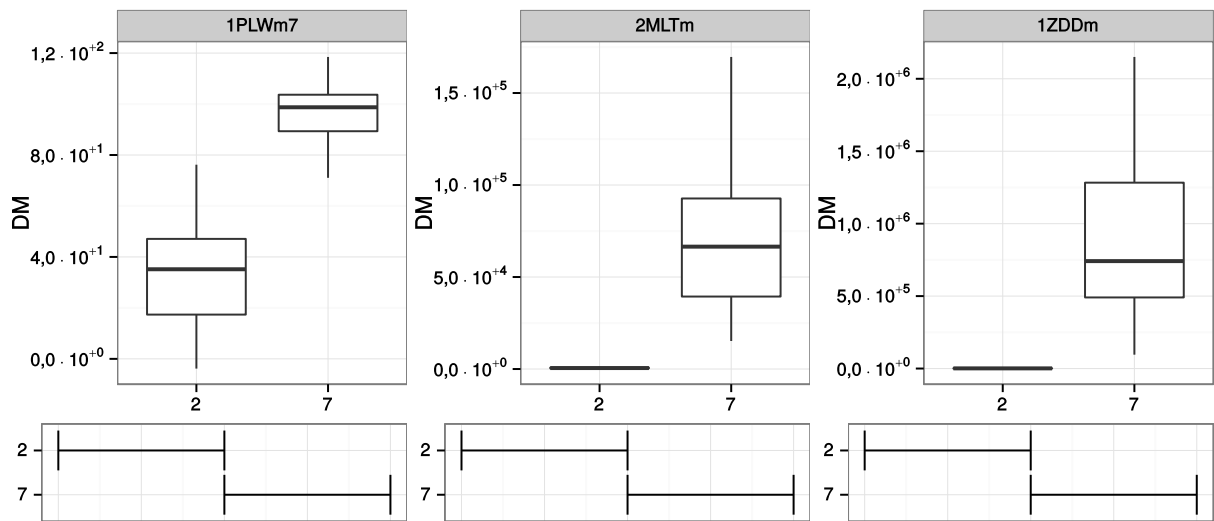


Figura 24: Resultados para PPEP com sete objetivos no algoritmo ELMOEA/D.

relação deteriora o desempenho do algoritmo com o aumento do número de objetivos pois muitas soluções tornam-se não dominadas (DEB; JAIN, 2014; YUAN *et al.*, 2014).

5 CONCLUSÕES

Este trabalho apresentou a proposta da plataforma ELMOEA/D para otimização de problemas multiobjetivo (MOPs) custosos. A proposta agrega algoritmos evolutivos multiobjetivo baseados em decomposição de objetivos e máquinas de aprendizado extremo para aproximar a avaliação custosa dos problemas.

Para a fundamentação dos conceitos utilizados na proposta, o Capítulo 2 apresentou a revisão dos temas otimização multiobjetivo, algoritmos evolutivos para otimização multiobjetivo (MOEAs), o algoritmo evolutivo para otimização multiobjetivo baseado em decomposição de objetivos original (MOEA/D) e duas variantes (MOEA/D-DE e MOEA/D-STM), seleção adaptativa de operadores, máquinas de aprendizado extremo e problema de predição de estruturas de proteínas (PPEP).

Em seguida, o Capítulo 3 trouxe uma breve revisão da literatura com trabalhos correlatos assim como detalhes da proposta (descrita passo-a-passo). Destacou-se que a plataforma ELMOEA/D gerencia dois arquivos de soluções, soluções avaliadas e soluções para treinamento do *surrogate*. Após a geração das soluções iniciais, até um determinado número de avaliações, o algoritmo ELMOEA/D itera nos passos: treinamento da *surrogate* ELM, otimização das funções (aproximadas) com uma variante do MOEA/D, seleção de soluções para avaliação, avaliação das soluções com a função custosa e atualização dos arquivos. Também foi apresentada a possibilidade de utilizar múltiplos *surrogates* e destacadas as diferenças com o algoritmo MOEA/D-RBF, que serviu de inspiração à proposta.

Os experimentos e resultados foram discutidos no Capítulo 4. Primeiramente, a metodologia dos testes foi descrita e os resultados foram apresentados em duas partes. Na primeira parte, a proposta foi testada com problemas *benchmark*, a melhor variante do algoritmo MOEA/D (MOEA/D original) foi escolhida e comparada ao algoritmo MOEA/D-RBF e ao algoritmo MOEA/D sem *surrogates*. Na segunda parte, a plataforma ELMOEA/D foi aplicada no PPEP, os parâmetros foram ajustados e algumas propriedades do PPEP foram discutidas com o auxílio dos resultados para diferentes parâmetros nos problemas *benchmarks* (Apêndice A). Em seguida,

a proposta configurada com os melhores parâmetros foi comparada ao algoritmo MOEA/D-DE sem *surrogates*. Por fim, os resultados para uma formulação alternativa do PPEP, com sete objetivos foram apresentados.

Os resultados da proposta nos problemas *benchmark* foram promissores. A plataforma ELMOEA/D supera o algoritmo MOEA/D sem *surrogates* em poucas avaliações, demonstrando que a utilização de *surrogates* pode ser vantajosa em MOPs custosos. Ainda, utilizando parâmetros semelhantes, a plataforma ELMOEA/D foi comparada ao algoritmo MOEA/D-RBF e apresentou melhores resultados na maioria das instâncias. Contudo, a investigação de problemas com múltiplas frentes de Pareto locais (como ZDT4) ainda é necessária, pois nenhum dos algoritmos comparados foi capaz de convergir e encontrar soluções Pareto-ótimas em poucas avaliações.

Os testes com *benchmarks* mostraram que a utilização de ELMs em MOEAs assistidos por *surrogates* traz bons resultados. Certamente, a investigação de diferentes variantes das redes ELM neste cenário é interessante (como ELMs ponderadas e ELMs incrementais (HUANG *et al.*, 2015)). Além disso, a escolha do parâmetro regularizador C da ELM, realizada antes dos experimentos, pode ser incorporada ao algoritmo, pois o estudo da parametrização automática dos *surrogates* ao longo da busca também é relevante em estudos mais recentes (ROSALES-PEREZ *et al.*, 2013).

O processo experimental conduzido ao longo da pesquisa permitiu ao final a obtenção de algumas conclusões. É mais importante, abriu várias linhas de investigação para trabalhos futuros. Entre elas pode-se destacar a verificação de algumas hipóteses:

- a utilização de poucas iterações e populações pequenas na otimização das funções (aproximadas) pelo MOEA/D parece melhorar o desempenho em problemas com múltiplas frentes de Pareto locais;
- a função de agregação Tchebycheff normalizada produz melhores resultados em problemas em que os intervalos de valores de cada objetivo diferem;
- como o cálculo dos valores de recompensa nas estratégias de seleção adaptativa de operadores pode ser feita em MOEAs assistidos por *surrogates*;
- a utilização de múltiplos modelos *surrogates* melhora o desempenho de MOPs com frente de Pareto descontínua.

A proposta não foi capaz de superar o algoritmo MOEA/D-DE sem *surrogates* em todos os casos no PPEP. Porém, a aplicação de MOEAs assistidos por *surrogates* no PPEP é inovadora,

sendo este um dos objetivos deste trabalho. Acredita-se que os resultados apresentados serviram como passos iniciais dessa abordagem para trabalhos futuros envolvendo:

- testes com os parâmetros que não foram ajustados, como o número de soluções selecionadas para avaliação (N_S), tamanho do arquivo de treinamento (N_T), número de neurônios ocultos na rede ELM (N_H) entre outros;
- investigação de diferentes estratégias de normalização das entradas e saídas para treinamento das redes ELM;
- utilização de técnicas alternativas para funções de agregação (como PBI invertido (SATO, 2014)) e distribuições de pesos (como a distribuição adaptativa (QI *et al.*, 2014) e baseada em ângulos (LIU *et al.*, 2014)), e quais as consequências na otimização das funções aproximadas e na manutenção do arquivo de treinamento;
- Estudo mais aprofundado sobre o PPEP, utilizando tomadores de decisão diferentes que avaliem melhor o compromisso entre objetivos, bem como os extremos da aproximação da fronteira de Pareto;
- Testes com diferentes proteínas, principalmente instâncias com mais variáveis e maiores demandas computacionais.

Apesar da aplicação da proposta ELMOEA/D não gerar bons resultados para a formulação com sete objetivos, a investigação deste tipo de formulação deverá ser continuada em trabalhos futuros. Como apresentado na Seção 3.2.2, o treinamento de *surrogates* ELM pode ser feito de maneira eficiente em relação ao número de objetivos. Porém MOPs com muitos objetivos ($N_O > 4$) certamente trazem novos desafios para MOEAs (ISHIBUCHI *et al.*, 2008) e a exploração de estratégias especializadas (como NSGA-III (DEB; JAIN, 2014)) também deverá ser foco de pesquisa futura.

REFERÊNCIAS

- ANFINSEN, C. B. *et al.* The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. **Proceedings of the National Academy of Sciences of the United States of America**, v. 47, p. 1309 – 1314, 1961.
- AUER, P.; CESA-BIANCHI, N.; FISCHER, P. Finite-time analysis of the multiarmed bandit problem. **Machine Learning**, Kluwer Academic Publishers, Hingham, MA, USA, v. 47, n. 2-3, p. 235 – 256, 2002.
- BARTLETT, P. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. **IEEE Transactions on Information Theory**, v. 44, n. 2, p. 525 – 536, 1998.
- BERMAN, H. M. *et al.* The protein data bank. **Nucleic Acids Research**, v. 28, n. 1, p. 235 – 242, 2000.
- BLEULER, S. *et al.* PISA — a platform and programming language independent interface for search algorithms. In: FONSECA, C. M. *et al.* (Ed.). **Evolutionary Multi-Criterion Optimization (EMO 2003)**. Berlin: Springer, 2003. p. 494 – 508.
- BUJNICKI, J. **Prediction of Protein Structures, Functions, and Interactions**. [S.l.]: Wiley, 2008.
- BURKE, E. K. *et al.* Hyper-heuristics: a survey of the state of the art. **Journal of the Operational Research Society**, Palgrave Macmillan, v. 64, n. 12, p. 1695 – 1724, 2013.
- COELLO, C. A. C.; LAMONT, G. B.; VELDHUIZEN, D. A. V. **Evolutionary Algorithms for Solving Multi-Objective Problems**. 2. ed. USA: Springer, 2007.
- CONOVER, W. J. **Practical Nonparametric Statistics**. 3. ed. [S.l.]: Wiley, 1999.
- CUTELLO, V.; NARZISI, G.; NICOSIA, G. A multi-objective evolutionary approach to the protein structure prediction problem. **Journal of The Royal Society Interface**, v. 3, n. 6, p. 139 – 51, 2006.
- DEB, K. **Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction**. [S.l.], 2011.
- DEB, K.; AGRAWAL, R. B. Simulated Binary Crossover for Continuous Search Space. **Complex Systems**, v. 9, n. 2, p. 115 – 148, 1995.
- DEB, K. *et al.* A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimisation: NSGA-II. In: **Proceedings of the 6th International Conference on Parallel Problem Solving from Nature**. London, UK: Springer-Verlag, 2000. p. 849 – 858.
- DEB, K.; GOYAL, M. A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. **Computer Science and Informatics**, v. 26, n. 4, p. 30 – 45, 1996.

- DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. **IEEE Transactions on Evolutionary Computation**, v. 18, n. 4, p. 577 – 601, 2014.
- DEMMELE, J. W. **Applied Numerical Linear Algebra**. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1997.
- EMMERICH, M.; NAUJOKS, B. Metamodel assisted multiobjective optimisation strategies and their application in airfoil design. In: PARMEE, I. (Ed.). **Adaptive Computing in Design and Manufacture VI**. [S.l.]: Springer London, 2004. p. 249 – 260.
- ENGELBRECHT, A. **Computational Intelligence: An Introduction**. New York, NY, USA: Halsted Press, 2002.
- GOLDBERG, D. E. Probability matching, the magnitude of reinforcement, and classifier system bidding. **Machine Learning**, Kluwer Academic Publishers, Hingham, MA, USA, v. 5, n. 4, p. 407 – 425, 1990.
- GONG, W. *et al.* Adaptive strategy selection in differential evolution for numerical optimization: An empirical study. **Information Sciences**, v. 181, n. 24, p. 5364 – 5386, 2011.
- GUO, L.; HAO, J. hua; LIU, M. An incremental extreme learning machine for online sequential learning problems. **Neurocomputing**, v. 128, n. 0, p. 50 – 58, 2014.
- HAO, J. hua; LIU, M. A surrogate modelling approach combined with differential evolution for solving bottleneck stage scheduling problems. In: **World Automation Congress (WAC), 2014**. [S.l.: s.n.], 2014. p. 120 – 124.
- HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. 2. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.
- HAYKIN, S. O. **Neural Networks and Learning Machines**. 3. ed. [S.l.]: Prentice Hall, 2008.
- HUANG, G. *et al.* Trends in extreme learning machines: A review. **Neural Networks**, v. 61, n. 0, p. 32 – 48, 2015.
- HUANG, G.-B. *et al.* Extreme learning machine for regression and multiclass classification. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics**, v. 42, n. 2, p. 513 – 529, 2012.
- HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: a new learning scheme of feedforward neural networks. In: **Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on**. [S.l.: s.n.], 2004. v. 2, p. 985 – 990.
- ISHIBUCHI, H.; TSUKAMOTO, N.; NOJIMA, Y. Evolutionary many-objective optimization: A short review. In: **IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)**. [S.l.: s.n.], 2008. p. 2419 – 2426.
- JIN, Y. A comprehensive survey of fitness approximation in evolutionary computation. **Soft Computing Journal**, v. 9, n. 1, p. 3 – 12, 2005.
- JIN, Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. **Swarm and Evolutionary Computation**, v. 1, n. 2, p. 61 – 70, 2011.

JONES, D.; SCHONLAU, M.; WELCH, W. Efficient global optimization of expensive black-box functions. **Journal of Global Optimization**, Kluwer Academic Publishers, v. 13, n. 4, p. 455 – 492, 1998.

KABSCH, W.; SANDER, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. **Biopolymers**, Wiley Subscription Services, Inc., A Wiley Company, v. 22, n. 12, p. 2577 – 2637, 1983.

KNOWLES, J.; CORNE, D. The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation. In: **Proceedings of the 1999 Congress on Evolutionary Computation (CEC 99)**. [S.l.: s.n.], 1999. v. 1, p. 98 – 105.

KNOWLES, J.; HUGHES, E. Multiobjective optimization on a budget of 250 evaluations. In: COELLO, C. C.; AGUIRRE, A. H.; ZITZLER, E. (Ed.). **Evolutionary Multi-Criterion Optimization**. [S.l.]: Springer Berlin Heidelberg, 2005. v. 3410, p. 176 – 190.

LI, H.; ZHANG, Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. **IEEE Transactions on Evolutionary Computation**, IEEE Press, Piscataway, NJ, USA, v. 13, n. 2, p. 284 – 302, 2009.

LI, K. *et al.* Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. **IEEE Transactions on Evolutionary Computation**, v. 18, n. 1, p. 114 – 130, 2014.

LI, K. *et al.* Stable matching based selection in evolutionary multiobjective optimization. **IEEE Transactions on Evolutionary Computation**, v. 18, n. 6, p. 909 – 923, 2014.

LIM, D. *et al.* Generalizing surrogate-assisted evolutionary computation. **IEEE Transactions on Evolutionary Computation**, v. 14, n. 3, p. 329 – 355, 2010.

LIM, D. *et al.* A study on metamodeling techniques, ensembles, and multi-surrogates in surrogate-assisted memetic algorithms. In: **Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation**. New York, NY, USA: ACM, 2007. p. 1288 – 1295.

LIU, H.-L.; GU, F.; ZHANG, Q. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. **IEEE Transactions on Evolutionary Computation**, v. 18, n. 3, p. 450 – 455, 2014.

LÓPEZ-IBÁÑEZ, M.; PAQUETE, L.; STÜTZLE, T. Exploratory analysis of stochastic local search algorithms in biobjective optimization. In: BARTZ-BEIELSTEIN, T. *et al.* (Ed.). **Experimental Methods for the Analysis of Optimization Algorithms**. [S.l.]: Springer, Berlin, Germany, 2010. p. 209 – 222.

MACKERELL, A. D.; BANAVALI, N.; FOLOPPE, N. Development and current status of the charmm force field for nucleic acids. **Biopolymers**, John Wiley & Sons, Inc., v. 56, n. 4, p. 257 – 265, 2000.

MARCOTTE, I. *et al.* A multidimensional 1h {NMR} investigation of the conformation of methionine-enkephalin in fast-tumbling bicelles. **Biophysical Journal**, v. 86, n. 3, p. 1587 – 1600, 2004.

- MARTÍNEZ, S. Z.; COELLO, C. A. C. MOEA/D Assisted by Rbf Networks for Expensive Multi-objective Optimization Problems. In: **Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference**. [S.l.: s.n.], 2013. p. 1405 – 1412.
- MCKAY, M. D.; BECKMAN, R. J.; CONOVER, W. J. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. **Technometrics**, Taylor & Francis, v. 21, p. 239 – 245, 1979.
- MIETTINEN, K. **Nonlinear Multiobjective Optimization**. [S.l.]: Kluwer Academic Publishers, 1999.
- NAIN, P.; DEB, K. Computationally effective search and optimization procedure using coarse to fine approximations. In: **The 2003 Congress on Evolutionary Computation (CEC '03)**. [S.l.: s.n.], 2003. v. 3, p. 2081 – 2088.
- NEUMAIER, A. Molecular modeling of proteins and mathematical prediction of protein structure. **SIAM Review**, v. 39, p. 407 – 460, 1997.
- NOCEDAL, J.; WRIGHT, S. J. **Numerical Optimization**. 2. ed. [S.l.]: Springer, 2006.
- PAVELSKI, L. *et al.* ELMOEA/D-DE: Extreme Learning Surrogate Models in Multi-objective Optimization based on Decomposition and Differential Evolution. In: **Proceedings of 2014 Brazilian Symposium on Intelligent Systems (BRACIS)**. Sao Paulo, Brazil: [s.n.], 2014. p. 318 – 323.
- PETSKO, G.; RINGE, D. **Protein Structure and Function**. [S.l.]: New Science Press, 2004.
- PONDER, J. W. Tinker: Software tools for molecular design. **Washington University School of Medicine, Saint Louis, MO**, v. 3, 2004.
- PONDER, J. W.; CASE, D. A. Force fields for protein simulations. **Advances in protein chemistry**, v. 66, p. 27 – 85, 2003.
- PONWEISER, W. *et al.* Multiobjective optimization on a limited budget of evaluations using model-assisted \mathcal{S} -metric selection. In: **Parallel Problem Solving from Nature – PPSN X**. [S.l.]: Springer Berlin Heidelberg, 2008. v. 5199, p. 784 – 794.
- QI, Y. *et al.* MOEA/D with Adaptive Weight Adjustment. **IEEE Transactions on Evolutionary Computation**, MIT Press, Cambridge, MA, USA, v. 22, n. 2, p. 231 – 264, 2014.
- ROSALES-PEREZ, A. *et al.* A hybrid surrogate-based approach for evolutionary multi-objective optimization. In: **IEEE Congress on Evolutionary Computation 2013 (CEC 2013)**. [S.l.: s.n.], 2013. p. 2548 – 2555.
- ROSALES-PEREZ, A. *et al.* Surrogate-assisted multi-objective model selection for support vector machines. **Neurocomputing**, v. 150, Part A, n. 0, p. 163 – 172, 2015.
- RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 2. ed. [S.l.]: Pearson Education, 2003.
- SATO, H. Inverted pbi in moea/d and its impact on the search performance on multi and many-objective optimization. In: **Proceedings of the 2014 Conference on Genetic and Evolutionary Computation**. New York, NY, USA: ACM, 2014. (GECCO '14), p. 645 – 652.

- SINDHYA, K. *et al.* A new hybrid mutation operator for multiobjective optimization with differential evolution. **Soft Computing**, v. 15, n. 10, p. 2041 – 2055, 2011.
- STORN, R.; PRICE, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, Kluwer Academic Publishers, v. 11, n. 4, p. 341 – 359, 1997.
- TRAMONTANO, A. **Protein structure prediction: concepts and applications**. [S.l.]: Wiley-VCH, 2006.
- TSIGELNY, I. F. **Protein Structure Prediction: Bioinformatic Approach**. [S.l.]: International University Line, 2002.
- WOLPERT, D.; MACREADY, W. No free lunch theorems for optimization. **Evolutionary Computation, IEEE Transactions on**, v. 1, n. 1, p. 67 – 82, 1997.
- XU, Y.; XU, D.; LIANG, J. **Computational methods for protein structure prediction and modeling**. [S.l.]: Springer, 2007.
- YUAN, Y.; XU, H.; WANG, B. An improved nsga-iii procedure for evolutionary many-objective optimization. In: **Proceedings of the 2014 Conference on Genetic and Evolutionary Computation**. New York, NY, USA: ACM, 2014. p. 661 – 668.
- ZHANG, Q.; LI, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. **IEEE Transactions on Evolutionary Computation**, v. 11, n. 6, p. 712 – 731, 2007.
- ZHANG, Q.; LIU, W.; LI, H. The performance of a new version of moea/d on cec09 unconstrained mop test instances. In: **Proceedings of the Eleventh Conference on Congress on Evolutionary Computation**. Piscataway, NJ, USA: IEEE Press, 2009. p. 203 – 208.
- ZHANG, Q. *et al.* Expensive Multiobjective Optimization by MOEAD With Gaussian Process Model. **IEEE Transactions on Evolutionary Computation**, v. 14, n. 3, p. 456 – 474, 2010.
- ZHANG, Q.; SUGANTHAN, P. N. Final report on CEC'09 MOEA competition. In: **Congress on Evolutionary Computation 2009 (CEC 2009)**. [S.l.: s.n.], 2009.
- ZITZLER, E. **Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications**. Tese (Doutorado) — ETH Zurich, Switzerland, 1999.
- ZITZLER, E.; KÜNZLI, S. Indicator-based selection in multiobjective search. In: YAO, X. *et al.* (Ed.). **Parallel Problem Solving from Nature - PPSN VIII**. [S.l.]: Springer Berlin Heidelberg, 2004. v. 3242, p. 832 – 842.
- ZITZLER, E.; LAUMANN, M.; THIELE, L. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: **Evolutionary Methods for Design, Optimisation, and Control**. [S.l.]: CIMNE, Barcelona, Spain, 2002. p. 95 – 100.
- ZITZLER, E.; THIELE, L. Multiobjective optimization using evolutionary algorithms - a comparative case study. In: **Proceedings of the 5th International Conference on Parallel Problem Solving from Nature**. London, UK, UK: Springer-Verlag, 1998. p. 292 – 304.

**APÊNDICE A – RESULTADOS PARA DIFERENTES PARÂMETROS DO
ELMOEA/D EM PROBLEMAS *BENCHMARK***

Tabela 10: Resultados do algoritmo ELMOEA/D para diferentes parâmetros do MOEA/D: média e desvio padrão do indicador Hipervolume em configurações com 10 e 100 gerações, populações de tamanho 100 e 300

MOP	N_D	N_E	$G-N$		$G-N$	
			10-100	10-300	100-100	100-300
ZDT1	8	200	0.592248 ± 0.316031	0.101330 ± 0.139534	0.008242 ± 0.007103	0.004601 ± 0.000416
ZDT2	8	200	6.191827 ± 2.924856	1.453986 ± 2.619533	0.008142 ± 0.008664	0.010523 ± 0.010363
ZDT3	8	200	0.204897 ± 0.217157	0.124393 ± 0.181725	0.086935 ± 0.087116	0.052846 ± 0.056233
ZDT4	8	200	1315.607 ± 301.5307	1431.976 ± 382.3524	1589.008 ± 476.5875	1706.449 ± 372.2544
ZDT6	8	200	3.949470 ± 1.704386	1.980910 ± 1.798611	1.378078 ± 1.325589	0.814625 ± 1.161673
ZDT1	30	1000	0.477355 ± 0.212036	0.116486 ± 0.064877	0.002924 ± 0.001272	0.005104 ± 0.000949
ZDT2	30	1000	3.320891 ± 0.801907	0.008921 ± 0.004761	0.002188 ± 0.000661	0.004206 ± 0.001808
ZDT3	30	1000	0.375870 ± 0.359033	0.394275 ± 0.179158	0.567710 ± 0.297817	0.416961 ± 0.320079
ZDT4	10	1000	512.2393 ± 154.5280	874.8654 ± 228.6498	962.0532 ± 243.6109	1073.470 ± 229.1957
ZDT6	10	1000	1.138797 ± 0.794090	0.379655 ± 0.399253	0.069092 ± 0.090269	0.248175 ± 0.369773
ZDT1	60	2000	0.943117 ± 0.248051	0.367024 ± 0.167301	0.013995 ± 0.011508	0.006974 ± 0.003693
ZDT2	60	2000	3.891287 ± 0.448165	0.595051 ± 0.960229	0.013292 ± 0.004276	0.013782 ± 0.004571
ZDT3	60	2000	2.001505 ± 0.996189	0.912117 ± 0.374199	0.567816 ± 0.224876	0.562457 ± 0.245252
ZDT4	20	2000	2499.666 ± 0.000000	2499.666 ± 0.000000	2499.666 ± 0.000000	2499.666 ± 0.000000
ZDT6	20	2000	3.434972 ± 1.648600	1.562008 ± 1.098470	0.085379 ± 0.162666	0.058311 ± 0.133820

Tabela 11: Resultados do algoritmo ELMOEA/D para diferentes funções de agregação (F): média e desvio padrão do indicador Hipervolume em configurações com as abordagens PBI, TCHE e NTCHE

MOP	N_D	N_E	PBI	TCHE	NTCHE
ZDT1	8	200	0.008242 \pm 0.007103	0.030706 \pm 0.026084	0.060359 \pm 0.077300
ZDT2	8	200	0.008142 \pm 0.008664	0.005557 \pm 0.001510	0.144100 \pm 0.676934
ZDT3	8	200	0.086935 \pm 0.087116	0.107108 \pm 0.119102	0.183465 \pm 0.173129
ZDT4	8	200	1589.008 \pm 476.5875	1338.838 \pm 398.7166	1347.466 \pm 320.9186
ZDT6	8	200	1.378078 \pm 1.325589	2.784434 \pm 1.328877	5.110166 \pm 5.744279
ZDT1	30	1000	0.002924 \pm 0.001272	0.061913 \pm 0.036516	0.101541 \pm 0.042779
ZDT2	30	1000	0.002188 \pm 0.000661	0.001961 \pm 0.000414	0.005081 \pm 0.001131
ZDT3	30	1000	0.567710 \pm 0.297817	0.544301 \pm 0.342354	0.306693 \pm 0.213969
ZDT4	10	1000	962.0532 \pm 243.6109	1049.958 \pm 252.7405	929.3314 \pm 185.5196
ZDT6	10	1000	0.069092 \pm 0.090269	0.465002 \pm 0.233623	0.693930 \pm 0.324360
ZDT1	60	2000	0.013995 \pm 0.011508	0.058508 \pm 0.025910	0.040246 \pm 0.017885
ZDT2	60	2000	0.013292 \pm 0.004276	0.014183 \pm 0.020926	0.021955 \pm 0.013764
ZDT3	60	2000	0.567816 \pm 0.224876	0.568267 \pm 0.350642	0.710358 \pm 0.366714
ZDT4	20	2000	2499.666 \pm 0.000000	2499.666 \pm 0.000000	2499.666 \pm 0.000000
ZDT6	20	2000	0.085379 \pm 0.162666	0.481027 \pm 0.142406	2.443552 \pm 0.861811

Tabela 12: Resultados do algoritmo ELMOEA/D para diferentes operadores de cruzamento: média e desvio padrão do indicador Hipervolume em configurações com os operadores *DE/rand/1/bin*, *DE/rand/2/bin* e *DE/nonlinear*, e com as estratégias de seleção adaptativa de operadores PM e FRRMAB.

MOP	N_D	N_E	<i>DE/rand/1/bin</i>	<i>DE/rand/2/bin</i>	<i>DE/nonlinear</i>	PM	FRRMAB
ZDT1	8	200	0.017021 ± 0.013657	0.018302 ± 0.021456	0.004479 ± 0.000631	0.010678 ± 0.007765	0.019785 ± 0.013445
ZDT2	8	200	0.260897 ± 1.372395	0.007077 ± 0.003476	0.006280 ± 0.004864	0.015388 ± 0.026793	0.008611 ± 0.009456
ZDT3	8	200	0.085149 ± 0.079588	0.058066 ± 0.067821	0.086881 ± 0.073408	0.065481 ± 0.067376	0.059145 ± 0.081280
ZDT4	8	200	1651.200 ± 442.5923	1555.865 ± 397.0465	1803.913 ± 433.2004	1524.947 ± 566.9385	1627.792 ± 387.1151
ZDT6	8	200	1.076550 ± 1.416454	1.325797 ± 1.426388	1.929979 ± 1.723075	0.928299 ± 1.149508	1.354136 ± 1.623852
ZDT1	30	1000	0.002438 ± 0.000775	0.004286 ± 0.001009	0.003585 ± 0.001156	0.002497 ± 0.000790	0.002626 ± 0.000768
ZDT2	30	1000	0.092768 ± 0.476499	0.843845 ± 1.300888	0.002805 ± 0.001138	0.297738 ± 0.902468	0.277371 ± 0.835212
ZDT3	30	1000	0.623263 ± 0.344905	0.763368 ± 0.369452	0.512379 ± 0.307190	0.669665 ± 0.271471	0.621818 ± 0.285381
ZDT4	10	1000	1160.359 ± 174.8438	996.7678 ± 275.0706	1019.307 ± 281.9202	1069.665 ± 224.7541	1066.132 ± 292.9085
ZDT6	10	1000	0.089514 ± 0.100290	0.141780 ± 0.122219	0.420314 ± 0.249895	0.168627 ± 0.281163	0.137600 ± 0.234692
ZDT1	60	2000	0.003361 ± 0.001131	0.006156 ± 0.002546	0.006764 ± 0.003222	0.003149 ± 0.002426	0.003271 ± 0.001357
ZDT2	60	2000	0.023481 ± 0.011762	0.031240 ± 0.042609	0.023881 ± 0.006277	0.021064 ± 0.006387	0.029559 ± 0.042249
ZDT3	60	2000	1.019881 ± 0.499922	0.897877 ± 0.335389	0.671584 ± 0.232066	0.869077 ± 0.393796	0.972468 ± 0.398811
ZDT4	20	2000	2499.666 ± 0.000000	2499.666 ± 0.000000	2499.666 ± 0.000000	2499.666 ± 0.000000	2499.666 ± 0.000000
ZDT6	20	2000	0.156851 ± 0.190041	0.612638 ± 1.022196	1.198104 ± 0.487933	0.146760 ± 0.181278	0.131485 ± 0.197196

Tabela 13: Resultados do algoritmo ELMOEA/D para múltiplos *surrogates*: média e desvio padrão do indicador Hipervolume em configurações com $N_a = 1, 2, 5$ e 10 *surrogates*.

MOP	N_D	N_E	1	2	5	10
ZDT1	8	200	0.068695 \pm 0.109304	0.605349 \pm 0.435503	0.530859 \pm 0.445238	0.724251 \pm 0.644524
ZDT2	8	200	0.009811 \pm 0.009524	2.349222 \pm 3.103628	0.659645 \pm 1.618587	0.525683 \pm 1.668072
ZDT3	8	200	0.141011 \pm 0.135476	5.129151 \pm 5.025086	5.436921 \pm 3.771783	4.957685 \pm 4.793827
ZDT4	8	200	2478.130 \pm 82.02529	863.2730 \pm 459.4325	944.1571 \pm 437.9186	944.4730 \pm 398.1829
ZDT6	8	200	4.552570 \pm 5.668380	2.347831 \pm 1.248905	2.341141 \pm 1.271826	2.686234 \pm 1.223640
ZDT1	30	1000	0.116881 \pm 0.050357	0.022407 \pm 0.012870	0.024405 \pm 0.011681	0.030396 \pm 0.020061
ZDT2	30	1000	0.005272 \pm 0.002004	0.982239 \pm 0.938083	1.092764 \pm 0.926816	1.012572 \pm 0.835222
ZDT3	30	1000	0.417374 \pm 0.231857	0.169771 \pm 0.085546	0.167819 \pm 0.102707	0.171492 \pm 0.106009
ZDT4	10	1000	2495.966 \pm 19.92518	1041.047 \pm 274.8820	963.5268 \pm 314.3512	763.8862 \pm 255.3493
ZDT6	10	1000	0.658431 \pm 0.299676	0.819755 \pm 0.486190	0.744520 \pm 0.431643	0.758314 \pm 0.619389
ZDT1	60	2000	0.042082 \pm 0.015805	0.317765 \pm 0.171306	0.333231 \pm 0.177424	0.277476 \pm 0.137668
ZDT2	60	2000	0.021882 \pm 0.014143	3.036296 \pm 1.288487	2.798506 \pm 1.352679	2.583169 \pm 1.263209
ZDT3	60	2000	0.693557 \pm 0.291659	1.244871 \pm 0.624428	1.488142 \pm 0.974191	1.354676 \pm 0.641095
ZDT4	20	2000	2499.666 \pm 0.000000	2499.666 \pm 0.000000	2499.666 \pm 0.000000	2499.666 \pm 0.000000
ZDT6	20	2000	2.424004 \pm 0.738397	9.295397 \pm 5.784262	11.40097 \pm 6.278659	8.527328 \pm 7.144282