



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS REITOR JOÃO DAVID FERREIRA LIMA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Thiago Henrique Lemos Fonseca

**MÉTODO EVOLUTIVO BASEADO EM ESPAÇOS DE BUSCA DE BAIXA  
DIMENSIONALIDADE PARA PROBLEMAS DE OTIMIZAÇÃO CONTÍNUA EM  
LARGA ESCALA**

Florianópolis, Santa Catarina – Brasil  
2023



Thiago Henrique Lemos Fonseca

**MÉTODO EVOLUTIVO BASEADO EM ESPAÇOS DE BUSCA DE BAIXA  
DIMENSIONALIDADE PARA PROBLEMAS DE OTIMIZAÇÃO CONTÍNUA EM  
LARGA ESCALA**

Tese submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

**Orientador(a):** Sílvia Modesto Nassar, Dra.

Florianópolis, Santa Catarina – Brasil

2023

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Fonseca, Thiago Henrique Lemos  
Método Evolutivo baseado em Espaços de Busca de Baixa  
Dimensionalidade para Problemas de Otimização em Larga  
Escala / Thiago Henrique Lemos Fonseca ; orientadora,  
Sílvia Modesto Nassar, 2023.  
128 p.

Tese (doutorado) - Universidade Federal de Santa  
Catarina, Centro Tecnológico, Programa de Pós-Graduação em  
Ciência da Computação, Florianópolis, 2023.

Inclui referências.

1. Ciência da Computação. 2. Modelo de Misturas  
Gaussianas. 3. Redução de Dimensionalidade. 4. Evolução  
Diferencial. I. Nassar, Sílvia Modesto. II. Universidade  
Federal de Santa Catarina. Programa de Pós-Graduação em  
Ciência da Computação. III. Título.

Thiago Henrique Lemos Fonseca

**MÉTODO EVOLUTIVO BASEADO EM ESPAÇOS DE BUSCA DE BAIXA  
DIMENSIONALIDADE PARA PROBLEMAS DE OTIMIZAÇÃO CONTÍNUA EM  
LARGA ESCALA**

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

**Júlio César Nievola, Dr.**

Pontifícia Universidade Católica do Paraná – PUCPR

**Alexandre César Muniz de Oliveira, Dr.**

Universidade Federal do Maranhão – UFMA

**Mauro Roisenberg, Dr.**

Universidade Federal de Santa Catarina – UFSC

**Elder Rizzon Santos, Dr.**

Universidade Federal de Santa Catarina – UFSC

Certificamos que esta é a versão original e final do trabalho de conclusão que foi julgado adequado para obtenção do título de doutor em Ciência da Computação.

---

Coordenação do Programa de  
Pós-Graduação em Ciência da  
Computação - UFSC

---

**Silvia Modesto Nassar, Dra.**

Orientador(a)

Universidade Federal de Santa  
Catarina – UFSC

Florianópolis, Santa Catarina – Brasil, 2023.

*Este trabalho é dedicado às minhas tias e avós,  
pelas quais o sonho do Doutorado se manteve vivo em mim.*

## AGRADECIMENTOS

Minha primeira e mais profunda gratidão é a Deus, que me concedeu resiliência e perseverança diante dos inúmeros desafios encontrados durante o desenvolvimento desta pesquisa. Em seguida, expresso minha sincera gratidão à minha esposa, Andressa Fontenele Rozental Cordeiro. Sua presença constante, apoio inabalável e sacrifícios pessoais foram pilares fundamentais em minha jornada. Sem o seu amor e comprometimento, a concretização desta tese não seria viável.

Devo um agradecimento especial à minha família. Em particular, à minha mãe e avós: Raquel Santana Lemos, Maria de Nazaré Santos Fonseca e Terezinha de Jesus Reis Santana. Embora tenham partido antes de testemunhar a conclusão deste doutorado, suas memórias, sonhos e anseios compartilhados comigo foram a chama que manteve minha determinação viva. Além disso, sou eternamente grato às minhas tias, Rosimar da Conceição Fonseca Ramos, Marta Maria Farias Silva, Rosane Maria Fonseca Ramos e Isabel Cristina Reis Santana. Elas desempenharam um papel fundamental na minha formação, inculcando em mim a importância da educação e do compromisso com os estudos.

Agradeço profundamente à minha amiga e orientadora, Professora Doutora Sílvia Modesto Nassar. Sua dedicação, paciência e expertise foram essenciais para o desenvolvimento e finalização desta pesquisa. Ela não apenas me guiou academicamente, mas também me inspirou com sua paixão e compromisso com a excelência. Estou imensamente grato por ter tido a oportunidade de aprender e crescer sob sua orientação e por ela ter compartilhado generosamente seu vasto conhecimento comigo. Também gostaria de expressar minha sincera gratidão ao professor Doutor Alexandre César Muniz de Oliveira. Seus conselhos ajudaram com aprimoramento do método proposto, tornando-se uma força motriz para as inovações e melhorias implementadas.

Em reconhecimento à Polytechnique de Montreal e à equipe do *Laboratoire en Intelligence des données - LID*, expresso minha profunda gratidão por me acolherem durante meu período de Doutorado Sanchuíche. Uma menção especial é dedicada ao professor Doutor Bruno Agard, meu supervisor externo no Canadá. A riqueza de seus conselhos e a profundidade de suas análises foram cruciais para o refinamento do método proposto nesta tese e para a consecução dos artigos dela derivados.

Por fim, expresso minha gratidão ao Programa de Pós-graduação em Ciência da Computação da UFSC e à CAPES pelo apoio financeiro através da bolsa de doutorado. Esse suporte foi essencial para a realização deste trabalho e para minha formação acadêmica.

*“Eu faço as máquinas mais humanas para que os humanos possam ser menos máquina.”*

Autor desconhecido

*“É preciso ter um caos dentro de si para dar à luz uma estrela cintilante.”*

Nietzsche

## RESUMO

A Otimização Global em Larga Escala (LSGO) tem sido um campo ativo de pesquisa, em parte devido às suas aplicações em áreas de ponta, como Deep Learning, Big Data e problemas complexos do mundo real, como criptografia de imagens e gerenciamento de tráfego em tempo real. A alta dimensionalidade apresenta um desafio significativo na resolução de problemas de otimização. A maldição da dimensionalidade refere-se à crescente dificuldade de encontrar soluções ótimas à medida que o número de dimensões do problema aumenta, tornando a questão desafiadora. Para abordar isso, esta tese propõe um método evolutivo denominado LSMDE (Evolução Diferencial baseada em Modelagem de Espaço de Baixa Dimensionalidade). O método LSMDE utiliza redução de dimensionalidade por meio da Decomposição em Valor Singular para construir um espaço de busca de baixa dimensionalidade a partir das soluções candidatas geradas por um algoritmo de evolução diferencial híbrido denominado GM-SHADE. Esse algoritmo incorpora um modelo de mistura gaussiana para melhor explorar o espaço de busca reduzido e mitigar a perda de informação ao mapear soluções de alta dimensionalidade para soluções de baixa dimensionalidade. Adicionalmente, o método proposto não exige conhecimento prévio da topologia do espaço de busca, tornando-o adaptável a diferentes problemas de LSGO. Esta tese buscou comparar o método proposto com as principais abordagens da literatura nos *benchmarks* mais reconhecidos para alta dimensionalidade, usando os critérios definidos pelo *IEEE CEC Special Sessions and Competitions on Large-Scale Global Optimization*. Para analisar a significância das diferenças observadas, foi utilizado o teste estatístico de Kruskal-Wallis com  $\alpha$  igual 0,05. Experimentos realizados indicam a superioridade do LSMDE em diferentes características de espaços de busca, especialmente para funções parcialmente separáveis. Também se observou uma melhor performance do LSMDE em condições em que o número de avaliações da função objetivo é restrito, facilitando sua utilização em ambientes onde os recursos computacionais são limitados. Além disso, esta tese também realizou um teste de escalabilidade ao comparar o desempenho do LSMDE e seus principais competidores à medida que a dimensionalidade do problema aumenta. Para essa tarefa, utilizou-se a suíte de teste *bbob-largescale*. Os resultados demonstram a robustez do LSMDE ao aumento da dimensionalidade, alcançando uma taxa de acerto ao valor-alvo entre 40% e 80%.

**Palavras-chaves:** Modelo de Misturas Gaussianas. Evolução Diferencial. Redução de Dimensionalidade.

## ABSTRACT

Large-Scale Global Optimization (LSGO) has been an active research field, partly due to its applications in cutting-edge areas such as Deep Learning, Big Data, and real-world complex problems like image encryption and real-time traffic management. High dimensionality presents a significant challenge in solving optimization problems. The curse of dimensionality refers to the increasing difficulty of finding optimal solutions as the number of problem dimensions increases, making the issue challenging. In response to this challenge, this thesis proposes an evolutionary method named LSMDE (Low-Dimensional Space Modeling-based Differential Evolution). LSMDE uses dimensionality reduction through Singular Value Decomposition to construct a low-dimensionality search space from the candidate solutions generated by a hybrid differential evolution algorithm called GM-SHADE. This algorithm incorporates a Gaussian mixture model to better explore the reduced search space and mitigate information loss when mapping high-dimensionality solutions to low-dimensionality ones. Additionally, the proposed method does not require prior knowledge of the search space topology, making it adaptable to various LSGO problems. This thesis sought to compare the proposed method with the main approaches in the literature on the most recognized benchmarks for high dimensionality, using the criteria defined by the IEEE CEC Special Sessions and Competitions on Large-Scale Global Optimization. To analyze the significance of the observed differences, the Kruskal-Wallis statistical test was used with  $\alpha$  set at 0.05. Experiments conducted indicate LSMDE's superiority in different search space characteristics, especially for partially separable functions. A better performance of LSMDE was also observed in conditions where the number of objective function evaluations is limited, facilitating its use in environments where computational resources are constrained. Furthermore, this thesis also conducted a scalability test by comparing LSMDE's performance and its main competitors as the problem's dimensionality increases. For this task, the bbob-largescale test suite was used. The results demonstrate LSMDE's robustness to increasing dimensionality, achieving a success rate to the target value between 40% and 80%.

**Palavras-chaves:** Gaussian Mixture Models. Differential Evolution. Dimensionality reduction.

## LISTA DE FIGURAS

Figura 1	– <i>Estratégia de otimização das principais abordagens de resolução de problemas LSGO</i> . . . . .	33
Figura 2	– Ilustração do processo de mutação . . . . .	47
Figura 3	– Exemplo de um processo de cruzamento . . . . .	48
Figura 4	– Ilustração visual (em 1D) do espaço de busca $[a, b]$ . O $c$ é o centro do intervalo, $x$ é a solução candidata e $\hat{x}$ é o candidato-oposto. . . . .	49
Figura 5	– A apresentação visual em 1D (a) e 2D (c) do esquema de segmentação baseado no centro, onde $x_1 \in [a_1, b_1]$ e $x_2 \in [a_2, b_2]$ . . . . .	51
Figura 6	– Ilustração em 1D ((a) e (b)) e 2D (c) da região de busca oposta modular, onde $x_1 \in [a_1, b_1]$ e $x_2 \in [a_2, b_2]$ . . . . .	52
Figura 7	– Ilustração em 1D ((a) e (b)) e 2D (c) da região de busca quase-oposta, onde $x_1 \in [a_1, b_1]$ e $x_2 \in [a_2, b_2]$ . . . . .	53
Figura 8	– (a) Gráficos da distribuição gaussiana univariada dada pela Equação 25 para vários parâmetros $\mu$ e $\sigma$ , e (b) contornos da distribuição gaussiana multivariada (2D) para vários parâmetros de $\mu$ e $\Sigma$ . . . . .	62
Figura 9	– Ilustração do algoritmo EM para duas componentes gaussianas. . . . .	65
Figura 10	– Ilustração do processo de <i>Clustering</i> em um espaço reduzido $\Psi$ 2D utilizando $k$ -means . . . . .	69
Figura 11	– Ilustração de clusterização em um espaço reduzido $\Psi$ 2D utilizando modelo de misturas gaussianas . . . . .	70
Figura 12	– Ilustração de clusterização em um espaço reduzido $\Psi$ 2D utilizando modelo de misturas gaussianas . . . . .	71
Figura 13	– Visão de alto nível do método LSMDE. As etapas principais estão destacadas em cinza . . . . .	76
Figura 14	– Decomposição de matriz . . . . .	79
Figura 15	– Ilustração de uma representação de baixa dimensão de uma matriz aleatória $P$ de $d$ dimensões para uma matriz $\Psi$ de $k$ dimensões. . . . .	80
Figura 16	– Projeção de um novo indivíduo de alta dimensão $x_i$ através da aproximação de baixa ordem $\phi(x_i \in P)$ . . . . .	80
Figura 17	– Cálculo do lema de Johnson-Lindenstraus para diferentes tamanhos de população do LSMDE . . . . .	81
Figura 18	– Ilustração do processo de transformação de espaço $\mathbb{R}^d \rightarrow \mathbb{R}^k$ . . . . .	82

Figura 19	– Modelos de mistura gaussiana encontrando regiões promissoras no espaço de busca de baixa dimensionalidade da função Alpine N.1. O número de <i>clusters</i> é automaticamente inferido. Neste caso, o número de <i>clusters</i> inicialmente definido (população $\Psi^t$ ) não corresponde à verdadeira distribuição geradora das próximas populações; portanto, o modelo de mistura gaussiana ajusta o número de <i>clusters</i> de acordo com a distribuição atual da população.	86
Figura 20	– Ilustração de uma transformação de espaço de uma Gaussiana promissora em $\Psi$ para uma região promissora em $\tilde{P}$ .	87
Figura 21	– A gaussiana promissora $\theta$ representada pelo seu centro $\mu$ é capaz de gerar uma amostra de $n$ soluções promissoras de baixa dimensionalidade. Essas soluções podem ser transformadas para o espaço de original de alta dimensionalidade através da função de transformação $\phi'$ .	88
Figura 22	– Pontuação média geral para 8 competidores com base no critério Formula 1 considerando os três experimentos. LSMDE tem o melhor desempenho geral.	99
Figura 23	– Pontuação de cada algoritmo concorrente com base no critério Formula 1 para $1, 2 \times 10^5$ avaliações	101
Figura 24	– Pontuação de cada algoritmo concorrente com base no critério Formula 1 para $6 \times 10^5$ avaliações	101
Figura 25	– Pontuação de cada algoritmo concorrente com base no critério Formula 1 para $3 \times 10^6$ avaliações	102
Figura 26	– Pontuação geral baseada no critério Formula 1 para (a) $1, 2 \times 10^5$ avaliações, (b) $6 \times 10^5$ avaliações e (c) $3 \times 10^6$ avaliações.	103
Figura 27	– Número das melhores funções parcialmente separáveis. LSMDE vs SHADE-ILS e LSMDE vs GL-SHADE de acordo com o teste <i>post-hoc</i> de Conover para $1, 2 \times 10^5$ , $6 \times 10^5$ e $3 \times 10^6$ avaliações.	106
Figura 28	– <i>Gráfico ECDF mostrando o desempenho do (a) LSMDE, (b) SHADE-ILS e (c) GL-SHADE, para diferentes proporções de número de avaliações/dimensões em todos os 24 problemas do suite de teste bbob-largescale.</i>	108
Figura 29	– Gráficos ECDF do LSMDE para as funções $f_3, f_6, f_8, f_9, f_{14}$ and $f_{15}$ no conjunto bbob-largescale, agregados em todas as instâncias de cada função.	109
Figura 30	– <i>ECDF para os três algoritmos selecionados, agregados em todas as 24 funções do conjunto bbob-largescale.</i>	110
Figura 31	– ECDF para o grupo de estruturas globais fracas funciona com 160-D, 320-D e 640-D.	111

## LISTA DE QUADROS

Quadro 1	–	Trabalhos selecionados . . . . .	30
Quadro 2	–	Algoritmos concorrentes . . . . .	42

## LISTA DE TABELAS

Tabela 1	–	Resumo das funções de <i>benchmark CEC'2013</i> . . . . .	94
Tabela 2	–	Parâmetros usados no LSMDE . . . . .	95
Tabela 3	–	Resumo de pontuação . . . . .	95
Tabela 4	–	Resumo da pontuação máxima alcançável por categoria de função	96
Tabela 5	–	Resumo da suite de teste <i>bbob-largescale</i> . . . . .	97
Tabela 6	–	Valores médios de aptidão para os algoritmos nas funções LSGO do CEC 2013 com $1,2 \times 10^5$ , $6 \times 10^5$ e $3 \times 10^6$ avaliações. . . . .	100
Tabela 7	–	Valores médios de aptidão para algoritmos com características semelhantes à proposta de tese nas funções LSGO do <i>CEC'2013</i> com $3 \times 10^6$ avaliações. . . . .	104
Tabela 8	–	Resultados do teste de Dunn, <i>p</i> -valores do LSMDE versus SHADE-ILS e GL-SHADE nas funções de referência CEC'2013 para 1000 dimensões. Os símbolos "+", "-" e "="denotam que o LSMDE é melhor que, pior que ou similar ao algoritmo comparado, respectivamente. . . . .	105
Tabela 9	–	Resultados do teste de Conover, <i>p</i> -valores de LSMDE versus SHADE-ILS e GL-SHADE nas funções de <i>benchmark CEC 2013</i> para 1000 dimensões. Os símbolos "+", "-" e "="indicam que o LSMDE é melhor, pior ou similar ao algoritmo comparado, respectivamente. . . . .	106

## LISTA DE ALGORITMOS

1	EM para Misturas Gaussianas . . . . .	64
2	SVD Adaptativo $\phi$ . . . . .	82
3	Inversa do SVD Adaptativo $\phi'$ . . . . .	82
4	LSMDE . . . . .	90

## LISTA DE ABREVIATURAS E SIGLAS

LSGO	Otimização Global em Larga Escala
LSMDE	Evolução Diferencial baseada em Modelagem de espaço de baixa dimensionalidade
GM-SHADE	Evolução Diferencial com adaptação de parâmetros baseado em histórico de sucesso e modelo de mistura gaussiana
SHADE	Evolução Diferencial com adaptação de parâmetros baseado em histórico de sucesso
SHADE-ILS	Evolução Diferencial com adaptação de parâmetros baseado em histórico de sucesso e busca local iterativa
GL-SHADE	Evolução Diferencial com adaptação de parâmetros usando busca local e global baseado em histórico de sucesso
DE	Evolução Diferencial
GMM	Modelos de mistura de gaussianas
SVD	Decomposição em Valores Singulares
CC	Coevolução Cooperativa
OBL	Aprendizado baseado em Oposição
JL	Lema de Johnson–Lindenstrauss
EM	Algoritmo de Maximização de Expectativa
MTS-LS1	Busca Local por Múltiplas trajetórias
L-BFGS-B	Algoritmo de Broyden–Fletcher–Goldfarb–Shanno com Memória Limitada
ECDF	Funções de Distribuição Cumulativa Empírica
FE	Avaliação de função
RR	Problema Reduzido Restrito
IR	Problema Irrestrito
R	Problema Restrito

## LISTA DE SÍMBOLOS

$\epsilon$	Fator de distorção na transformação de espaço
$\gamma(z_{nc})$	Probabilidade a posteriori de que o dado $x_n$ foi gerado pelo componente $c$
$\hat{x}$	Solução candidata oposta para o problema de otimização em alta dimensionalidade
$\Lambda$	Conjunto de todos os clusters
$\lambda$	Um cluster específico
$\mu$	Indivíduo central de um cluster
$\omega$	Parâmetro de concentração do processo de Dirichlet
$\phi$	Função de transformação de espaço
$\pi$	Coefficiente de mistura
$\Psi$	Espaço de busca de baixa dimensionalidade
$\psi$	Solução candidata para o problema de otimização em baixa dimensionalidade
$\psi^*$	Solução ótima global para o problema de otimização em baixa dimensionalidade
$\rho$	Limite de coeficiente de mistura
$\Sigma$	Variância de um cluster multidimensional
$\theta$	Parâmetros de formação da gaussiana
$\theta^*$	Parâmetros ótimos de uma gaussiana
$CR$	Taxa de cruzamento
$d$	Dimensão do problema de alta dimensionalidade
$F$	Fator de escala da mutação
$f$	Função objetivo
$k$	Dimensionalidade do problema reduzido
$n$	Número de indivíduos em uma população de soluções
$P$	Espaço de busca de alta dimensionalidade
$x$	Solução candidata para o problema de otimização em alta dimensionalidade

- $x^*$  Solução ótima global para o problema de otimização em alta dimensionalidade
- $x^{cb}$  Solução candidata oposta para o problema de otimização em alta dimensionalidade com segmentação de intervalo baseado no centro
- $x^{mo}$  Solução candidata oposta para o problema de otimização em alta dimensionalidade com segmentação de intervalo oposto modular
- $x^{qo}$  Solução candidata oposta para o problema de otimização em alta dimensionalidade com segmentação de intervalo quase oposto
- $z_n$  Variável aleatória categórica

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>19</b>
1.1	MOTIVAÇÃO E JUSTIFICATIVA	19
1.2	PROBLEMA DE PESQUISA	21
1.3	OBJETIVO GERAL	22
<b>1.3.1</b>	<b>Objetivos Específicos</b>	<b>22</b>
1.4	CONTRIBUIÇÕES E INEDITISMO	22
1.5	PUBLICAÇÕES & RECONHECIMENTOS	23
1.6	ADERÊNCIA AO PPGCC/LINHA DE PESQUISA	24
1.7	ORGANIZAÇÃO DO TRABALHO	24
<b>1.7.1</b>	<b>Capítulo 2: Revisão da Literatura</b>	<b>24</b>
<b>1.7.2</b>	<b>Capítulo 3: Fundamentação Teórica</b>	<b>25</b>
<b>1.7.3</b>	<b>Capítulo 4: Modelagem do espaço de busca de baixa dimensionalidade</b>	<b>25</b>
<b>1.7.4</b>	<b>Capítulo 5: Proposta de Tese</b>	<b>26</b>
<b>1.7.5</b>	<b>Capítulo 6: Resultados</b>	<b>26</b>
<b>1.7.6</b>	<b>Capítulo 7: Conclusões</b>	<b>27</b>
<b>2</b>	<b>REVISÃO DA LITERATURA</b>	<b>28</b>
2.1	ELEMENTOS DE BUSCA E SELEÇÃO	28
2.2	TRABALHOS RELACIONADOS	32
<b>2.2.1</b>	<b>Métodos baseados em Decomposição</b>	<b>32</b>
<b>2.2.2</b>	<b>Métodos Não-baseados em Decomposição</b>	<b>37</b>
<b>2.2.3</b>	<b>Estado da Arte</b>	<b>42</b>
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>44</b>
3.1	METAHEURÍSTICAS BASEADAS EM POPULAÇÃO	44
3.2	EVOLUÇÃO DIFERENCIAL	45
<b>3.2.1</b>	<b>Mutação</b>	<b>46</b>
<b>3.2.2</b>	<b>Cruzamento</b>	<b>47</b>
<b>3.2.3</b>	<b>Seleção</b>	<b>48</b>
3.3	APRENDIZADO BASEADO EM OPOSIÇÃO	48
<b>3.3.1</b>	<b>Segmentações do espaço de busca baseados em oposição</b>	<b>49</b>
3.3.1.1	Ponto central	50
3.3.1.2	Intervalo baseado no centro	50
3.3.1.3	Intervalo oposto modular	51
3.3.1.4	Intervalo quase oposto	52
3.4	REDUÇÃO DE DIMENSIONALIDADE BASEADO EM DECOMPOSIÇÃO DE VALOR SINGULAR	53

3.4.1	<b>Vetor singular</b> . . . . .	54
3.4.2	<b>Decomposição em Valores Singulares (SVD)</b> . . . . .	56
3.4.3	<b>Lema de Johnson-Lindenstrauss</b> . . . . .	57
3.5	MODELOS DE MISTURA . . . . .	58
3.5.1	<b>Modelo de Misturas Gaussianas</b> . . . . .	61
4	<b>MODELAGEM DE ESPAÇO DE BUSCA DE BAIXA DIMENSIONALIDADE</b> . . . . .	66
4.1	PRINCIPIOS DE TRANSFORMAÇÃO DE ESPAÇO . . . . .	66
4.2	PROBLEMAS REDUZIDOS VIA DECOMPOSIÇÃO DE VALOR SINGULAR . . . . .	68
4.3	TRANSFORMAÇÃO DE ESPAÇO . . . . .	68
4.4	MODELAGEM DE ESPAÇO DE BUSCA REDUZIDO $\Psi$ . . . . .	69
5	<b>PROPOSTA DA TESE</b> . . . . .	76
5.1	DESCRIÇÃO DO MÉTODO PROPOSTO (LSMDE) . . . . .	76
5.1.1	<b>Inicialização da População</b> . . . . .	77
5.1.2	<b>Transformação de espaço</b> . . . . .	78
5.1.3	<b>Otimização &amp; Agrupamento em <math>\Psi</math></b> . . . . .	83
5.1.4	<b>Busca Local</b> . . . . .	88
5.2	PROCEDIMENTOS METODOLÓGICOS . . . . .	91
5.2.1	<b>Análise de performance</b> . . . . .	93
5.2.2	<b>Análise de escalabilidade</b> . . . . .	96
6	<b>RESULTADOS</b> . . . . .	99
6.1	ANÁLISE DE PERFORMANCE DO LSMDE COMPARADO COM O ESTADO DA ARTE . . . . .	99
6.2	ANÁLISE DE PERFORMANCE DE ALGORITMOS SIMILARES AO MÉTODO PROPOSTO . . . . .	103
6.3	ANÁLISE ESTATÍSTICA DO LSMDE COMPARADA COM O ESTADO DA ARTE . . . . .	104
6.4	ANÁLISE DE ESCALABILIDADE DO LSMDE COMPARADA COM O ESTADO DA ARTE . . . . .	107
7	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	112
	<b>REFERÊNCIAS</b> . . . . .	114

# 1 INTRODUÇÃO

## 1.1 MOTIVAÇÃO E JUSTIFICATIVA

A área de Otimização Global em Larga Escala (*Large Scale Global Optimization* - LSGO) lida com problemas de otimização em alta dimensionalidade, ou seja, cujo número de variáveis de decisão é considerado alto (KIM; CHOI, 2020). Embora o limiar exato possa variar dependendo do contexto específico, muitos pesquisadores consideram que problemas com mais de 1.000 variáveis reais são de alta dimensionalidade (CHEN; TAN, 2023; LIU; CHENG *et al.*, 2023; LI; WANG *et al.*, 2023). No entanto, problemas com apenas algumas centenas de variáveis também podem ser considerados de alta dimensionalidade, se a estrutura do problema for complexa ou se requisitos computacionais necessários forem intensivos. Por exemplo, problemas de otimização global que envolvem redes neurais profundas ou grandes conjuntos de dados podem ser considerados de alta dimensionalidade, mesmo que o número absoluto de variáveis não seja extremamente alto, devido à complexidade inerente desses problemas (MOUSAVIRAD; RAHNAMAYAN, 2020).

O estudo de problemas LSGO é de grande relevância para a Ciência da Computação, não apenas pelo desenvolvimento de novos métodos evolutivos para esses problemas, mas também pelo incentivo às linhas de pesquisa intimamente ligadas à projetos computacionais em larga escala, como Sistemas Distribuídos e Algoritmos Paralelos (DE FALCO; DELLA CIOPPA; TRUNFIO, 2019). A aplicabilidade de LSGO em áreas atuais como *Deep Learning*, *Big Data*, entre outros, também é um fator que atrai o interesse pelo projeto de algoritmos cada vez mais eficientes. (OMIDVAR; LI, 2017; TABERNIK; SKOČAJ, 2019). Contudo, vários fatores tornam a resolução de problemas LSGO um grande desafio. Por exemplo, a avaliação de problemas em larga escala costuma ser computacionalmente custosa (LI, L. *et al.*, 2021). Este é frequentemente o caso em muitos problemas do mundo real, como encriptação de imagens (CHOWDHARY *et al.*, 2020; TAMANG *et al.*, 2021), gerenciamento de tráfego em tempo real (LUAN *et al.*, 2020), entre outros.

Outro fator que contribui para a dificuldade na resolução de problemas de Otimização em Larga Escala é a interação entre variáveis. Essa interação impede que as mesmas sejam otimizadas de forma independente para encontrar o ótimo global de uma função objetivo (LI; GUO *et al.*, 2021; DE FALCO; DELLA CIOPPA; TRUNFIO, 2019). Na literatura de Otimização Contínua, a interação entre variáveis é comumente referida como Não-separabilidade (JAMIL; YANG, 2013; CARAFFINI; NERI; PICINALI, 2014) e na literatura de Algoritmos Genéticos, esse fenômeno é denominado Epistasia ou Interação Gênica (CHEN, Y.-p. *et al.*, 2007; JAFARI *et al.*, 2019).

Para lidar com a Não-separabilidade, técnicas de Decomposição com Coevolução Cooperativa são frequentemente empregadas para dividir as variáveis de decisão do

problema em vários subproblemas, cada um contendo um subconjunto das dimensões originais (REN *et al.*, 2021; XUE *et al.*, 2020; DE FALCO; DELLA CIOPPA; TRUNFIO, 2019). No entanto, o particionamento das dimensões potencialmente reduz a cobertura do espaço de busca, uma vez que o resultado da otimização é a combinação de múltiplos subespaços otimizados. Além disso, a identificação do melhor agrupamento de dimensões exige avaliações de custo computacional mais elevado (DEL SER *et al.*, 2019; REN *et al.*, 2021; LI; GUO *et al.*, 2021; KOÇER; UYMAZ, 2021).

Para mitigar o problema da dependência de variáveis, Metaheurísticas Não-baseadas em Decomposição vem sendo desenvolvidas ao longo dos últimos anos (SCHOEN; TIGLI, 2021; JIAN; CHEN *et al.*, 2021; ZHANG; ZHAN; ZHANG, 2020; DE FALCO; DELLA CIOPPA; TRUNFIO, 2019). Essa abordagem consiste em aprimorar a exploração de espaços de busca de alta dimensionalidade através da criação de novos operadores evolutivos, métodos de inicialização, buscas locais especializadas e estratégias que otimizam o espaço multidimensional de maneira conjunta, sendo responsáveis pelo desenvolvimento de importantes algoritmos para problemas LSGO (PACHECO-DEL-MORAL; COELLO, 2020; MOLINA; LATORRE; HERRERA, 2018). Contudo, esses algoritmos ainda sofrem com o crescimento exponencial do espaço de busca e de ótimos locais devido ao aumento da dimensionalidade do problema (REN *et al.*, 2021; LI, L. *et al.*, 2021). Pesquisas recentes também sugerem que o balanceamento entre exploração e intensificação é prejudicado em espaços de alta dimensionalidade (KOCER; UYMAZ, 2018; BOLUFÉ-RÖHLER; CHEN; TAMAYO-VERA, 2019). Portanto, uma estratégia de busca mais eficiente é necessária para explorar regiões promissoras nesses espaços (KOCER; UYMAZ, 2018; BOLUFÉ-RÖHLER; CHEN; TAMAYO-VERA, 2019).

Bagattini, Schoen e Tigli (2018) e Bagattini, Schoen e Tigli (2019) demonstraram que é possível reduzir os efeitos da dimensionalidade através de uma otimização em um espaço de busca reduzido dimensionalmente. Os resultados, apesar de encorajadores, se basearam na análise prévia de características do espaço de busca para desenvolvimento de técnicas de redução de dimensionalidade específicas para cada tipo de espaço, não sendo possível estendê-las a problemas gerais de LSGO (SCHOEN; TIGLI, 2021).

Para avançar as pesquisas anteriores e desenvolver um método que não se baseie em suposições iniciais sobre características do espaço, esta tese propõe utilizar uma técnica de redução de dimensionalidade por Aprendizado de Máquina Não-Supervisionado para reduzir dinamicamente o espaço de busca gerado por uma metaheurística populacional. Foi escolhido a Evolução Diferencial (*Differential Evolution - DE*) como metaheurística devido aos excelentes resultados para problemas LSGO promovidos por suas variantes ao longo dos últimos anos (DE FALCO; DELLA CIOPPA; TRUNFIO, 2019; MOLINA; NESTERENKO; LATORRE, 2019; MAUČEC; BREST, 2019) e por permitir uma análise comparativa com os principais algoritmos para LSGO

disponíveis na literatura (muitos dos quais são variantes de DE).

## 1.2 PROBLEMA DE PESQUISA

Métodos evolutivos são abordagens poderosas para problemas de otimização complexos, senão impossíveis, de resolver por meio de métodos analíticos convencionais. No entanto, quando esses problemas apresentam alta dimensionalidade, o desempenho de muitos métodos evolutivos decai devido a diversos fatores, entre eles, o mais conhecido é a "maldição da dimensionalidade" (REN *et al.*, 2021; LI, L. *et al.*, 2021). Esse fenômeno se refere ao rápido aumento da complexidade e do custo computacional à medida que a dimensionalidade do problema aumenta.

Entre outros fatores que contribuem para a complexidade na resolução de problemas de alta dimensionalidade estão a não-separabilidade, a multimodalidade e o mal-condicionamento. A não-separabilidade implica que as variáveis do problema interagem entre si, impossibilitando a resolução do problema simplesmente resolvendo subproblemas definidos por cada variável individualmente (LI; GUO *et al.*, 2021; DE FALCO; DELLA CIOPPA; TRUNFIO, 2019). A multimodalidade, por outro lado, é a existência de vários ótimos locais, além do ótimo global, que podem desviar os algoritmos de encontrar a solução ideal. Por fim, o mal-condicionamento dos problemas, ou seja, a diferença na sensibilidade da função de aptidão ao variar diferentes direções, é outra dificuldade que pode dificultar a otimização em larga escala. Essa questão é especialmente problemática quando o mal-condicionamento muda dependendo da posição no espaço de busca ou quando o problema mal-condicionado também é não-separável.

Para contornar esse e outros fatores, técnicas de redução de dimensionalidade podem ser utilizadas. Essas técnicas transformam o espaço de busca de alta dimensão em um espaço de busca de dimensão inferior, tornando o problema mais gerenciável (BAGATTINI; SCHOEN; TIGLI, 2018, 2019). No entanto, a redução de dimensionalidade não é isenta de desafios. Técnicas de redução de dimensionalidade muito específicas, que levam em conta as características do espaço de busca, restringem a aplicabilidade do método a diferentes problemas LSGO (SCHOEN; TIGLI, 2021). Em contra partida, algumas técnicas de redução de dimensionalidade baseadas em aprendizado de máquina podem ser uma alternativa para o problema, contudo, uma transformação de espaço de tal natureza pode introduzir incertezas e distorcer a estrutura do problema de otimização, potencialmente prejudicando a capacidade dos algoritmos desenvolvidos de encontrar a solução ideal.

Nesse contexto, a questão de pesquisa que esta tese busca responder é: **"Como desenvolver um método evolutivo eficaz que aproveite as vantagens da redução de dimensionalidade para resolver problemas de otimização em larga escala, ao mesmo tempo que gerencie efetivamente a incerteza e a distorção introduzidas pela transformação do espaço de busca?"**

Para responder a esta questão, esta tese propõe investigar o desenvolvimento de um método evolutivo que utiliza uma técnica de redução de dimensionalidade adaptativa baseada em aprendizado de máquina para tornar a otimização mais eficiente e Modelos de Misturas Gaussianas (GMM) para gerenciar a incerteza resultante do processo anterior. Espera-se que o GMM seja capaz de mitigar os efeitos relacionados com a perda de informação inerente ao processo de transformação de espaço de modo a viabilizar a exploração em um espaço de busca de baixa dimensionalidade.

### 1.3 OBJETIVO GERAL

Propor um método evolutivo baseado em Redução de Dimensionalidade e Modelos de Misturas Gaussianas para exploração eficiente de espaços de busca de problemas de Otimização Global em Larga Escala (LSGO).

#### 1.3.1 Objetivos Específicos

Os objetivos específicos desta tese são:

- Analisar de forma crítica as abordagens, metodologias, e avanços significativos no campo de Problemas de Otimização Global em Grande Escala (LSGO) publicados na literatura científica nos últimos dez anos, com o intuito de identificar tendências atuais, lacunas de pesquisa e oportunidades para futuros estudos.
- Investigar o desenvolvimento de uma técnica de redução de dimensionalidade adaptativa que possa ser acoplada a diferentes metaheurísticas populacionais em um processo de exploração de espaços de busca.
- Modelar o espaço de busca de baixa dimensionalidade através de modelos de misturas gaussianas afim de mitigar os efeitos ocasionados pela transformação de espaço.
- Desenvolver um método evolutivo que possa ser aplicável para diferentes metaheurísticas populacionais sob diferentes espaços de busca.
- Avaliar o método proposto.

### 1.4 CONTRIBUIÇÕES E INEDITISMO

Esta tese apresenta um método de Evolução Diferencial baseado em Espaços de Baixa Dimensionalidade (LSMDE) para abordagens não-baseadas em decomposição que utiliza uma técnica de redução de dimensionalidade adaptativa para construir um espaço de busca de baixa dimensionalidade a partir das características da população atual de soluções candidatas geradas por uma metaheurística populacional geral. O

método apresentado não requer conhecimento prévio sobre o espaço de busca para explorá-lo. Modelos de Mistura Gaussiana (GMM) são usados em conjunto com a metaheurística para promover a busca por soluções ótimas no espaço reduzido. O GMM mitiga o erro no mapeamento de indivíduos de alta dimensão para indivíduos de baixa dimensão por agrupamento suave usando distribuições de probabilidade. Entre as principais contribuições desta pesquisa encontram-se:

- Uma técnica de redução dimensionalidade adaptativa que pode ser utilizada por diferentes metaheurísticas baseadas em população.
- O desenvolvimento de um método evolutivo que permite o aprendizado a partir da população atual de soluções candidatas sem necessidade de conhecimento prévio sobre as características do espaço de busca, tornando-o mais flexível e aplicável a uma variedade de problemas LSGO.
- Uma investigação detalhada e crítica dos diversos algoritmos voltados para problemas LSGO que emergiram no curso da última década.
- O desenvolvimento do LSMDE, um algoritmo baseado no método proposto.
- Além disso, este trabalho inova pela sua abordagem interdisciplinar, ao combinar técnicas tradicionalmente usadas no aprendizado de máquina com metaheurísticas com o objetivo de resolver problemas de otimização global.

## 1.5 PUBLICAÇÕES & RECONHECIMENTOS

Até o momento, esta tese culminou em duas publicações relevantes em uma revista e evento da área, respectivamente:

- *Low-dimensional Space Modeling-based Differential Evolution for Large Scale Global Optimization Problems* - IEEE Transactions on Evolutionary Computation (2022) (**Qualis A1**) (FONSECA *et al.*, 2022).
- *Low-dimensional Space Modeling-based Differential Evolution: A scalability perspective on bbob-largescale suite* - International Work-Conference on Artificial Neural Networks (2023) (**Qualis B1**) (FONSECA *et al.*, 2023).

Além das publicações mencionadas, a pesquisa tema desta tese foi também reconhecida através de uma bolsa pelo programa CAPES-PRINT para um doutorado sanduíche na Polytechnique Montreal, Canadá. Uma parte significativa do desenvolvimento deste trabalho foi realizada no *Laboratoire en Intelligence des Données* (LID).

A pesquisa recebeu financiamento do Scale AI, uma iniciativa canadense que visa acelerar a adoção da inteligência artificial no setor industrial. Esses recursos foram

fundamentais para implementar uma versão do método proposto na empresa Logistik Inc, contribuindo assim para a aplicação prática dos resultados obtidos.

Adicionalmente, os frutos deste trabalho proporcionaram um financiamento para participação no evento AI WEEK 2022, organizado pelo Alberta Machine Intelligence Institute, proporcionando uma valiosa oportunidade de compartilhar e discutir os avanços da pesquisa com outros profissionais da área.

## 1.6 ADERÊNCIA AO PPGCC/LINHA DE PESQUISA

Esta tese está em consonância com a Linha de Pesquisa de Inteligência Computacional do Programa de Pós-Graduação em Ciência da Computação da UFSC em duas vertentes principais: A modelagem da proposta e a relevância do problema a ser resolvido.

- **Modelagem da proposta:** o desenvolvimento de um método que utiliza técnicas de Computação Evolutiva e Aprendizado de Máquina configura o uso inovador das abordagens previstas na Linha de Pesquisa de Inteligência Artificial.
- **Resolução do problema:** Problemas de Otimização em Larga Escala configuram uma área de importância crescente devido sua complexidade de resolução e impacto em diferentes tecnologias utilizadas na sociedade atual.

## 1.7 ORGANIZAÇÃO DO TRABALHO

O restante deste documento está estruturado da seguinte forma:

### 1.7.1 Capítulo 2: Revisão da Literatura

Neste capítulo, efetua-se uma revisão sistemática e abrangente da literatura pertinente ao campo de problemas de otimização em Larga-Escala (LSGO). O segmento inicial proporciona uma visão geral do estado atual da arte, seguido por uma análise de trabalhos anteriores relevantes para o objeto de investigação. Esta análise contempla:

- **Critérios de Seleção:** São elucidados os critérios adotados na seleção dos trabalhos que compõem a revisão, assegurando que tal escolha seja representativa e pertinente ao escopo da pesquisa em curso.
- **Metodologias e Técnicas:** Discutem-se em detalhe as metodologias e técnicas utilizadas nos trabalhos considerados na revisão, com enfoque nas suas vantagens e limitações.
- **Identificação de Lacunas:** Com base na revisão conduzida, identificam-se lacunas no conhecimento atual que fundamentam a necessidade da investigação empreendida.

A revisão da literatura estabelece não apenas o alicerce teórico para a pesquisa, mas também serve como indicativo para áreas que requerem estudo adicional, justificando assim o ineditismo e o impacto potencial do trabalho em desenvolvimento.

### 1.7.2 Capítulo 3: Fundamentação Teórica

Este capítulo é organizado em quatro subseções, cada uma abordando um pilar teórico essencial para a compreensão do método proposto:

- **Metaheurísticas Populacionais:** Dado que o método proposto emprega uma metaheurística populacional como mecanismo de busca, são discutidas as estratégias algorítmicas para otimização que utilizam um conjunto de soluções candidatas. Esta subseção tem como objetivo fornecer uma visão abrangente da técnica envolvida.
- **Aprendizado por Oposição:** A incorporação de um mecanismo de aprendizado por oposição é uma estratégia utilizada para enriquecer a diversidade da população inicial de soluções candidatas. Assim, são elucidados o conceito e as aplicações práticas do aprendizado por oposição no domínio da otimização.
- **Redução de Dimensionalidade por Decomposição de Valor Singular (SVD):** A mitigação da maldição da dimensionalidade é abordada por meio da redução da dimensionalidade do espaço de busca, para a qual o método SVD é utilizado. Esta subseção visa esclarecer a relevância e eficácia do SVD em contextos de grandes conjuntos de dados.
- **Modelos de Mistura Gaussiana:** Após a redução da dimensionalidade, torna-se crucial modelar de forma apropriada o espaço de busca reduzido. Para essa tarefa, modelos de mistura são considerados. Nesta subseção são discutidos os aspectos matemáticos desses modelos e sua importância na estrutura geral da pesquisa.

Ao abordar esses quatro pilares teóricos, esta seção tem como objetivo estabelecer uma base sólida para a compreensão, concepção e avaliação crítica do método proposto.

### 1.7.3 Capítulo 4: Modelagem do espaço de busca de baixa dimensionalidade

Este capítulo apresenta os conceitos matemáticos necessários para a interpretabilidade do espaço de busca de baixa dimensionalidade. Os conceitos matemáticos decorrentes da seção de fundamentação teórica são adaptados para permitir a exploração de um problema de otimização global. Os tópicos centrais abordados incluem:

- **Princípios da transformação de espaço:** Através de um exemplo ilustrativo, apresenta-se uma problemática de otimização e como é possível reduzir sua dimensionalidade. Esta abordagem permite a parametrização e a busca de soluções em um espaço reduzido dimensionalmente.
- **Problemas reduzidos via Decomposição de Valor Singular:** Explora-se a técnica de redução de dimensionalidade Decomposição de Valor Singular (SVD) para transformar um subespaço de alta dimensionalidade em um espaço de baixa dimensionalidade.
- **Modelagem de espaço de busca reduzido:** Aprofunda-se na utilização de modelos de mistura para representar o espaço de busca reduzido. Aspectos matemáticos como a Equação de Maximização de Verossimilhança e a atualização de parâmetros são detalhados. Em particular, a utilização do Algoritmo de maximização de expectativa (EM) e sua convergência são abordados.

#### 1.7.4 Capítulo 5: Proposta de Tese

Neste capítulo, detalha-se a proposta central da tese, focada na otimização em larga escala e na avaliação do desempenho de método proposto. Os tópicos centrais abordados são:

- **Descrição do método:** É apresentada a visão geral do LSMDE, destacando cada uma das principais etapas do processo. Em seguida, cada etapa é detalhada para a compreensão da proposta de tese.
- **Procedimentos metodológicos:** Apresenta-se os princípios norteadores para os experimentos realizados sobre o método proposto e seus principais competidores. Em especial, detalha-se os *benchmarks* e os critérios de avaliação utilizados na pesquisa

#### 1.7.5 Capítulo 6: Resultados

Este capítulo apresenta os principais resultados decorrentes da pesquisa. Os resultados são divididos em três categorias principais:

- **Análise de performance do LSMDE comparado com o estado da arte:** Apresentam-se as análises de desempenho do método proposto em comparação as principais abordagens da literatura segundo o *IEEE CEC Special Sessions and Competitions on Large-Scale Global Optimization*.
- **Análise de performance de algoritmos similares ao método proposto:** Apresentam-se as análises de desempenho do método proposto em comparação a abordagens similares, ou seja, que utilizam Aprendizado por Oposição ou

Redução de Dimensionalidade ou Modelos de Mistura Gaussiana, entre outros aspectos.

- **Análise estatística:** Apresentam-se as análises estatísticas que mostram o desempenho superior do LSMDE para diferentes características de espaço de busca quando comparado com as melhores abordagens dos experimentos anteriores.
- **Análise de escalabilidade:** Apresenta-se as análises que mostram que o LSMDE é robusto com relação ao aumento de dimensionalidade do problema para diferentes características de espaço de busca.

### 1.7.6 Capítulo 7: Conclusões

Neste último capítulo, articula-se os principais achados da pesquisa, refletindo sobre suas implicações, limitações e potenciais contribuições para o campo de otimização em Larga-Escala. A estrutura do capítulo se desdobra da seguinte maneira:

- **Síntese dos Resultados:** Resgatam-se os principais pontos da tese, consolidando as descobertas e enfatizando sua relevância no contexto mais amplo da literatura revisada.
- **Contribuições da Pesquisa:** Destacam-se as inovações e distinções que esse trabalho trouxe ao campo de estudo, considerando tanto aspectos teóricos quanto práticos.
- **Limitações e Desafios:** Com humildade acadêmica, apresenta-se aqui os limites do nosso estudo, ponderando as áreas que poderiam ser melhoradas ou que não foram abrangidas na presente pesquisa.
- **Recomendações para Pesquisas Futuras:** Baseado nas descobertas e nos desafios identificados, propõe-se direções promissoras para futuras investigações. Este segmento tem como objetivo inspirar e orientar pesquisadores que desejem aprofundar ou expandir o trabalho aqui iniciado.

## 2 REVISÃO DA LITERATURA

Este capítulo tem por objetivo apresentar as principais contribuições para Problemas de Otimização Global em Larga Escala - *Large-Scale Global Optimization Problems (LSGO)* na última década, bem como o detalhamento do processo de análise realizado para a construção do *corpus* de investigação e os instrumentos de busca e seleção aplicados.

### 2.1 ELEMENTOS DE BUSCA E SELEÇÃO

Para construção do *corpus* de investigação é necessário definir quais bases eletrônicas indexam os artigos mais relevantes produzidos dentro da área de pesquisa. As principais bases que indexam pesquisas na área de Ciência da Computação são (NEIVA; SILVA, 2016):

- Scopus
- IEEE Xplore
- ScienceDirect
- Springer
- ACM

Sobre a base Scopus, pontua-se que a busca por palavras-chave foi removida no final de 2020, mantendo apenas a busca por autor, periódico e área geral de pesquisa. Tais critérios permitem uma busca generalista, dificultando o direcionamento dos trabalhos encontrados para o foco de pesquisa desejado, além de diferir dos critérios de seleção utilizados nas demais bases. Logo, optou-se por retirar a Scopus da composição do *corpus* deste trabalho. Espera-se que trabalhos relevantes indexados na base Scopus estejam também indexados em outras bases aqui investigadas, de modo a mitigar os potenciais efeitos prejudiciais causados pela sua eliminação.

Definidas as bases, a próxima etapa a ser realizada é a construção da *string de busca*. A *string* deve conter a essência da pesquisa desenvolvida e refletir os aspectos específicos que permitam selecionar apenas os trabalhos que possam ser devidamente comparados e avaliados com a proposta de tese apresentada. A partir de um processo de construção e refinamento, a *string* desenvolvida para essa tese foi definida como:

("Large Scale Global Optimization" **OR** "Large Scale Optimization")

**AND**

("Uncertainty" **OR** "Dimensionality Reduction" **OR** "CEC'2013" **OR** "bbob-largescale")

As duas primeiras palavras-chave, *Large Scale Global Optimization* e *Large Scale Optimization*, representam a nomenclatura corrente utilizada para otimização em alta dimensionalidade ou Larga Escala (LSGO). Logo, a composição dessas palavras-chave retorna todo o histórico de trabalhos realizados na área de LSGO.

É importante que dentro desse grupo, sejam identificados os trabalhos que foram desenvolvidos utilizando princípios semelhantes à abordagem da tese. Logo, o operador Booleado AND aplicado às duas palavras-chave seguintes limita a busca aos trabalhos de LSGO que utilizaram algum tipo de modelagem de incerteza e/ou redução de dimensionalidade em seu escopo.

As últimas duas palavras-chave adicionam, ao *corpus* existente, trabalhos que foram testados nos principais *benchmarks* para a área de Otimização em Larga Escala: O *CEC'2013*, inicialmente apresentado no *CEC'2013 Special Session and Competition on Large-Scale Global Optimization* (LI, X. *et al.*, 2013) e sendo o mais recente *benchmark* usado até os dias atuais sob a revisão do *IEEE Task Force on Large-Scale Global Optimization*; e a suite de teste *bbob-largescale*, uma extensão do conjunto BBOB (Black-Box Optimization Benchmarking) original, que é comumente usado para avaliar o desempenho de algoritmos de otimização em alta dimensionalidade sob a perspectiva de escalabilidade (VARELAS *et al.*, 2020).

O *benchmark CEC'2013* possui um conjunto de 15 funções de teste para LSGO com diferentes características e níveis de complexidade. Enquanto a suite de teste *bbob-largescale* apresenta 24 funções com novas características em comparação às apresentadas pelo *CEC'2013*. Juntos, os dois *benchmarks* englobam uma grande variedade de espaços de busca e problemas de otimização contínua em larga escala. Mais detalhes sobre a composição desses *benchmark* e características dos espaços dos espaços de busca contidos neles são discutidos no capítulo 5 desta tese.

Verificou-se que os trabalhos de apresentação de métodos considerados estado da arte em LSGO estão englobados nestas palavras-chave. Entre outras vantagens, os *CEC'2013* e *bbob-largescale* permitem comparar os algoritmos derivados desses métodos de acordo com os mesmos padrões de avaliação e performance.

Em resumo, a *string de busca* completa adiciona ao *corpus* já composto de trabalhos sobre LSGO que utilizaram modelagem de incerteza e/ou técnicas de redução de dimensionalidade, os trabalhos sobre LSGO avaliados nos principais *benchmarks* que abrangem o estado da arte para o problema.

Foram considerados pesquisas publicadas em periódicos e eventos de Ciência da Computação na área de Inteligência Artificial e Otimização nos últimos dez anos, divididos em IEEE Xplore (56), ACM (136), Springer (379) e ScienceDirect (239). É importante ressaltar que um mesmo artigo pode estar indexado em mais de uma base.

A seleção dos artigos finais deu-se principalmente através da leitura dos resumos, salvo os casos em que as informações não estavam plenamente descritas e precisou-se verificar a secção de metodologia. Trabalhos não relacionados com o tema,

duplicados ou que utilizaram *benchmarks* anteriores aos *CEC'2013* e *bbob-largescale* não foram considerados, com exceção daqueles que utilizaram métodos semelhantes à abordagem proposta nesta tese ou serviram de base para algoritmos competitivos considerados na *IEEE Task Force on Large-Scale Global Optimization*.

O Quadro 1 representa os 61 principais trabalhos na área de LSGO dos últimos dez anos. Alguns trabalhos de 2012 foram incorporados a análise devido sua importância para o desenvolvimento da área de LSGO, com métodos que serviram de inspiração para vários trabalhos mais recentes ou que ainda se mantêm competitivos até os dias atuais. Os trabalhos no Quadro 1 foram selecionados para compor o *corpus* de investigação desta tese e categorizados de acordo com o uso ou não de uma estratégia baseada na decomposição do espaço de busca. A subsecção seguinte apresenta os detalhamento de algumas dessas pesquisas.

Quadro 1 – Trabalhos selecionados

Artigo	Decomposição	Não-Decomposição
Li, Wang <i>et al.</i> (2023)		✓
Chen e Tan (2023)	✓	
Rodrigues <i>et al.</i> (2023)		✓
Fonseca <i>et al.</i> (2022)		✓
Song, Wang e Cai (2022)		✓
Mai Sun <i>et al.</i> (2022)	✓	
Hiba, Rahnamayan <i>et al.</i> (2022)		✓
Li, Guo <i>et al.</i> (2021)		✓
Feng <i>et al.</i> (2021)		✓
Schoen e Tigli (2021)		✓
Ren <i>et al.</i> (2021)	✓	
Jian, Chen <i>et al.</i> (2021)		✓
Lin Li <i>et al.</i> (2021)	✓	
Koçer e Uymaz (2021)	✓	
Yang <i>et al.</i> (2020)	✓	
Xue <i>et al.</i> (2020)	✓	
Pacheco-Del-Moral e Coello (2020)		✓
Ge, Zhao <i>et al.</i> (2020)	✓	
Zhang, Zhan e Zhang (2020)		✓
Ma e Bai (2020)		✓
Hanbo Deng <i>et al.</i> (2019)		✓
De Falco, Della Cioppa e Trunfio (2019)	✓	
Kazimipour, Omidvar <i>et al.</i> (2019)	✓	
Zhang, Gong <i>et al.</i> (2019)	✓	

Quadro 1 - Continuação

Artigo	Decomposição	Não-Decomposição
Chen, Sun e Palade (2019)	✓	
Hiba, Ibrahim e Rahnamayan (2019)		✓
Hiba, El-Abd e Rahnamayan (2019)		✓
Yuan Sun <i>et al.</i> (2019)	✓	
Sopov e Vakhnin (2019)	✓	
Dong, Wang e Zhou (2019)		✓
Hadi, Mohamed e Jambi (2019)	✓	
Bagattini, Schoen e Tigli (2019)		✓
Sanyang e Kaban (2019)		✓
Bagattini, Schoen e Tigli (2018)		✓
Molina, LaTorre e Herrera (2018)		✓
Maučec, Brest, Bošković <i>et al.</i> (2018)		✓
Wang, Liu <i>et al.</i> (2018)	✓	
Segredo <i>et al.</i> (2018)		✓
An Chen <i>et al.</i> (2018)	✓	
Mahdavi, Rahnamayan e Shiri (2018)	✓	
Marcelino <i>et al.</i> (2018)		✓
Mohapatra, Das e Roy (2017)		✓
Hiba, Mahdavi e Rahnamayan (2017)		✓
Liu, Wang <i>et al.</i> (2017)	✓	
Mahdavi, Rahnamayan e Shiri (2017)	✓	
Glorieux <i>et al.</i> (2017)	✓	
Kabán, Bootkrajang e Durrant (2016)		✓
Mahdavi, Rahnamayan e Deb (2016)	✓	
Liu, Guan <i>et al.</i> (2015)	✓	
Ge, Sun <i>et al.</i> (2015)	✓	
Molina e Herrera (2015)		✓
Wei, Wang e Zong (2014a)	✓	
Wei, Wang e Zong (2014b)	✓	
Kazimipour, Li e Qin (2014)		✓
LaTorre, Muelas e Pena (2013)		✓
Wei, Wang e Huo (2013)	✓	
Liu e Tang (2013)	✓	
Korošec e Šilc (2013)		✓
Fister, Žumer <i>et al.</i> (2012)		✓
LaTorre, Muelas e Pena (2012)		✓

Quadro 1 - Continuação

Artigo	Decomposição	Não-Decomposição
Zhang e Li (2012)	✓	

Fonte: Autor

## 2.2 TRABALHOS RELACIONADOS

Vários algoritmos específicos foram propostos para lidar com problemas de LSGO na última década. De acordo com [Jian, Zhan e Zhang \(2020\)](#), existem duas categorias principais de abordagens para esse tipo de problema: métodos de decomposição baseados em Coevolução Cooperativa (CC) e métodos não-baseados em decomposição. Métodos baseados em decomposição seguem a abordagem de dividir e conquistar, decompondo os problemas LSGO em vários subcomponentes de baixa dimensionalidade, enquanto que métodos não-baseados em decomposição focam em aprimorar a exploração de espaços de busca  $d$ -dimensionais através da criação de novos operadores evolutivos e buscas locais especializadas. Esta tese foca na investigação e desenvolvimento de um método evolutivo para LSGO que pode ser classificado como não-baseado em decomposição. Contudo, o LSMDE apresenta uma diferença significativa ao reduzir a dimensionalidade do espaço de busca como um todo visando reduzir a complexidade do problema. A Figura 1 ilustra a estratégia de otimização das duas abordagens principais e do método proposto. Um breve histórico das duas abordagens é importante para a visualização do contexto geral na área de Problemas de Otimização em Larga Escala. As subseções a seguir fornecem uma visão geral dos algoritmos desenvolvidos a partir de ambos os métodos no período analisado.

### 2.2.1 Métodos baseados em Decomposição

Por muitos anos, Coevolução Cooperativa (CC) tem sido considerada um método baseado em decomposição promissor para problemas LSGO. Contudo, a convergência prematura e o custo computacional ainda são fatores que dificultam a sua utilização. [Zhang e Li \(2012\)](#) propuseram uma alternativa para minimizar os impactos desses problemas através do uso de CC combinado com uma busca global. A abordagem era baseada em dois estágios, um estágio de exploração de subespaço e um estágio de exploração do espaço global. Denominado CCGS, o algoritmo de dois estágios apresentou resultados superiores aos algoritmos de referência até aquele momento. Posteriormente, [Wei, Wang e Huo \(2013\)](#) propuseram uma busca paralela em múltiplas regiões do espaço de busca auxiliada por uma função de suavização que elimina soluções locais ao longo do processo de otimização, mitigando o problema do custo computacional associado à essas estratégias. Mais recentemente, [De Falco, Della Cioppa e Trunfio \(2019\)](#) investigaram um método CC assistido por uma função substituta

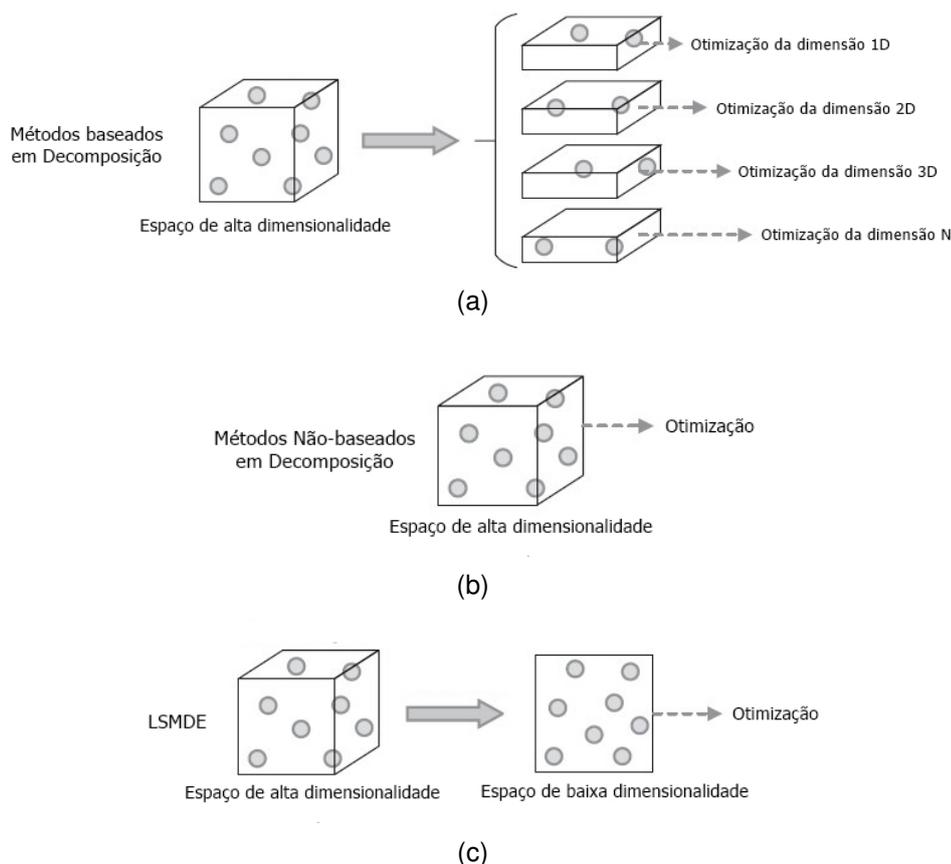


Figura 1 – Estratégia de otimização das principais abordagens de resolução de problemas LSGO

Fonte: Autor

(SACC), no qual substitutos de aptidão são explorados dentro dos subcomponentes de baixa dimensão resultantes da decomposição do problema. De acordo com os resultados, o SACC é capaz de diminuir significativamente a convergência prematura.

Uma estratégia intuitiva que visa melhorar o desempenho de algoritmos para LSGO é a inicialização de populações. Seguindo essa linha, [Mahdavi, Rahnamayan e Deb \(2016\)](#) propuseram uma Coevolução Cooperativa com estratégias de inicialização de populações baseadas em região central que obteve resultados promissores em funções não-separáveis de alta dimensionalidade, enquanto que [Glorieux et al. \(2017\)](#) apresentaram uma nova arquitetura de reinicialização no algoritmo denominado Coevolução Cooperativa Construtiva ( $C^3$ ). Um algoritmo de otimização integrado otimiza os subproblemas separadamente enquanto troca informações para co-adaptar as soluções para os subproblemas. Além disso,  $C^3$  inclui uma nova heurística construtiva que gera diferentes soluções viáveis para todo o problema e, assim, acelera a busca.

Uma das estratégias mais utilizadas em métodos baseados em decomposição é identificar relações entre variáveis para otimizá-las de maneira conjunta e diminuir a dimensionalidade do problema. Esse agrupamento pode ser aleatório ([SOPOV; VAKH-NIN, 2019](#)) ou de acordo com algum critério pré-definido ([WEI; WANG; ZONG, 2014a](#))

(LIU; WANG *et al.*, 2017). A busca por alternativas mais eficientes para a construção de subcomponentes menores levou os pesquisadores ao aprofundamento de novas técnicas de agrupamento de variáveis. Wei, Wang e Zong (2014b) propuseram dividir as variáveis do problema em dois grupos distintos: com interação e sem interação. A partir dessa divisão, é aplicado um algoritmo de evolução diferencial de características específicas para cada grupo. Resultados indicaram que a separação proposta alcançou resultados melhores que abordagens anteriores.

Para o melhor agrupamento de variáveis, é necessário identificar suas relações de dependência de maneira efetiva. Ge, Sun *et al.* (2015) propuseram um operador de busca rápida para capturar dependências. Além disso, os pesquisadores desenvolveram uma mutação entre grupos para melhorar a relação exploração e intensificação. Liu, Guan *et al.* (2015) e Wang, Liu *et al.* (2018) também desenvolveram estratégias para minimizar as limitações na detecção de dependências entre certas variáveis utilizando um agrupamento baseado em fórmula (FBG). No FBG, as variáveis podem ser agrupadas automaticamente em um número adequado de subcomponentes. Posteriormente, foi demonstrado que o agrupamento diferencial também tem um bom desempenho na decomposição de problemas LSGO, detectando efetivamente a interação entre as variáveis de decisão. An Chen *et al.* (2018) melhoraram a precisão da decomposição do agrupamento diferencial através de um algoritmo de definição de limiar adaptativo baseado em informações globais (GIAT), enquanto que Zhang, Gong *et al.* (2019) propuseram a criação de subcomponentes elitistas constituídos por variáveis superiores através de um método de estimativa baseado em informações históricas da melhor aptidão geral. Com base nessas informações de interação, uma estratégia de agrupamento dinâmico é conduzida para construir melhores subcomponentes. Seguindo na mesma linha, Lin Li *et al.* (2021) propuseram um novo algoritmo de decomposição fuzzy que agrupa as variáveis de acordo com seu grau de interação definido por funções de pertinência. Recentemente Mai Sun *et al.* (2022) propuseram uma otimização em larga escala com uma estratégia de alternância (LSEO-SS), um algoritmo que adota um método de agrupamento aleatório com dimensões variáveis para decompor um problema em larga escala em vários subproblemas sem interseção. Esse algoritmo usa modelos substitutos para aproximar o valor-alvo (aptidão), sem considerar as outras partes do problema, otimizando cada subproblema individualmente. Em seguida, as soluções para o problema geral são formadas combinando horizontalmente as soluções dos subproblemas, com uma sendo selecionada para avaliação objetiva com base em um critério de preenchimento dinâmico. Quando o algoritmo detecta que a população caiu em um ótimo local, um mecanismo de fuga forçada é acionado, reiniciando algumas dimensões da população.

A identificação de grupos de variáveis abre um novo foco de análise dentro da variedade de abordagens baseadas em decomposição. A maioria dos métodos baseados em decomposição, especialmente os modelos de Coevolução Cooperativa, assumem

que todos os subcomponentes são igualmente importantes e desafiadores. Como resultado, a alocação de recursos computacionais (em relação ao tamanho da população e ao número de iterações) ocorre uniformemente entre todos os subcomponentes. Na prática, entretanto, muitas aplicações são identificadas como tendo componentes com importância e dificuldade desiguais (OMIDVAR; LI; YAO, 2011). A fonte das contribuições desbalanceadas de componentes na qualidade da solução final pode ser qualquer combinação de dimensionalidade não uniforme, topologia do espaço de busca desigual, entre outros. Portanto, a Coevolução Cooperativa tradicional pode não funcionar de maneira adequada quando esse desequilíbrio existe. Com base nisso, Mahdavi, Rahnamayan e Shiri (2017) propuseram um *framework* de otimização multinível baseado no efeito de variáveis (MOFBVE) que calcula o efeito principal das variáveis usando um método de análise de sensibilidade auxiliado por uma técnica de clusterização k-means para construir grupos de variáveis com efeitos semelhantes no valor de aptidão. Os grupos construídos são classificados em ordem decrescente com base em sua contribuição no valor de aptidão e os grupos principais são selecionados como os níveis das variáveis importantes. Mahdavi, Rahnamayan e Shiri (2018) propuseram um método que aumenta gradativamente a importância dos subcomponentes através de um critério baseado em análise de sensibilidade visando gerenciar melhor os recursos computacionais, enquanto que Kazimipour, Omidvar *et al.* (2019) estenderam a Coevolução Cooperativa padrão para um novo método generalista capaz de aprender a contribuição de cada componente usando técnicas de bandidos multi-armados. O novo método aloca os recursos computacionais para cada componente de maneira proporcional às suas contribuições para melhorar o valor-alvo geral. Essa abordagem resulta em um uso mais econômico de recursos computacionais limitados. Chen, Sun e Palade (2019) propuseram uma otimização por enxame de partículas quânticas baseada em contribuição distribuída com diversidade controlada (DC-QPSO). De acordo com o nível de contribuição otimizada de cada subcomponente, os recursos computacionais são realocados automaticamente em cada etapa, garantindo que os subgrupos com maior contribuição obtenham mais recursos computacionais. Posteriormente, Yang *et al.* (2020) propuseram um algoritmo que pode especificar subpopulações de tamanhos desiguais para otimizar diferentes subcomponentes de variáveis. Denominado CCFR2, o algoritmo calcula a melhoria média do melhor valor-alvo geral por avaliação de aptidão como a contribuição de uma subpopulação. Um parâmetro de controle é adotado para equilibrar os efeitos das melhorias ao longo do processo de otimização.

Além de problemas desbalanceados, houve uma preocupação crescente nos últimos anos com problemas com sobreposição, ou seja, um caso particular de problemas não-separáveis onde algumas variáveis têm interação com outras variáveis em que não há uma única variável ou grupo de variáveis que possam ser identificados como um componente isolado. Yuan Sun *et al.* (2019) usaram um método de agrupamento diferencial recursivo (CC-RDG3) para decompor problemas sobrepostos, quebrando a

ligação de variáveis compartilhadas por vários componentes. Para avaliar a eficácia do método, os pesquisadores consideraram vários níveis de sobreposição. Os resultados experimentais mostraram que o método pode melhorar a capacidade de busca de um algoritmo de otimização via divisão e conquista quando o problema possui sobreposição. Posteriormente, [Xue et al. \(2020\)](#) propuseram um método promissor denominado agrupamento diferencial de *pool* único baseado em topologia (TSPDG) em que as interações das variáveis de decisão são identificadas de forma ordinal.

Uma alternativa interessante que vem para lidar com o aumento da dimensionalidade é o mapeamento do processo de otimização para um espaço reduzido alternativo. Essa estratégia pode ser usada por métodos baseados ou não em decomposição. Considerando a primeira classe, [Ge, Zhao et al. \(2020\)](#) propuseram um método generalista de Coevolução Cooperativa iterativa com evoluções em dois espaços (BICCA). No espaço de padrões, os padrões de interação de variáveis são continuamente avaliados pela CC. No espaço de busca, a Coevolução Cooperativa e a busca global são realizadas de forma adaptativa para obter uma melhor aptidão. Ao adotar evoluções e interações dentro de dois espaços, os padrões evoluem para fornecer melhores agrupamentos enquanto os indivíduos evoluem para alcançar uma melhor aptidão. A decomposição do problema é conduzida ao longo do processo de otimização. Experimentos no *benchmark CEC'2013* mostram que BICCA obtém desempenho competitivo em problemas de otimização com até 10.000 dimensões. Recentemente, [Ren et al. \(2021\)](#) propuseram uma abordagem de divisão e conquista por autoespaço (EDC). Diferente dos algoritmos baseados em divisão e conquista existentes que realizam decomposição e otimização no espaço de decisão original, o EDC primeiro estabelece um autoespaço conduzindo a decomposição de valor singular em um conjunto de soluções de alta qualidade selecionadas de gerações recentes. Em seguida, o algoritmo transforma o problema de otimização no autoespaço e, assim, enfraquece significativamente as dependências entre as variáveis correspondentes. Consequentemente, essas variáveis podem ser agrupadas de forma eficiente por uma estratégia aleatória simples e cada um dos subcomponentes resultantes pode ser tratado mais facilmente por um algoritmo evolutivo tradicional.

Alguns algoritmos baseados em decomposição também utilizaram hibridizações com métodos tradicionalmente não-baseados em decomposição. Como exemplo, [Hadi, Mohamed e Jambi \(2019\)](#) propuseram um SHADE com redução linear do tamanho da população e adaptação de semi-parâmetro (MLSHADE-SPA) onde as dimensões são divididas aleatoriamente em grupos, e cada grupo é resolvido separadamente. Posteriormente, um método de busca local adaptativo foi desenvolvido para o MLSHADE-SPA, alcançando resultados promissores em funções não separáveis e com sobreposição ([KOÇER; UYMAZ, 2021](#)).

### 2.2.2 Métodos Não-baseados em Decomposição

Apesar da melhora no desempenho computacional proporcionada pelas constantes atualizações de métodos baseados em Decomposição, tais ganhos obtidos geralmente não compensam o custo em termos de avaliações necessárias para detecção de dependências de variáveis. De acordo com [Del Ser et al. \(2019\)](#), essa abordagem não é considerada competitiva o suficiente em comparação com algoritmos especialmente projetados para LSGO. O interesse em mitigar as limitações dos métodos baseados em Decomposição levou à investigação de outras estratégias. Um trabalho interessante desenvolvido por [LaTorre, Muelas e Pena \(2013\)](#) descreve todo o processo de criação de um algoritmo híbrido competitivo específico para LSGO, desde o projeto experimental até a validação estatística final dos resultados, mostrando que um bom projeto experimental é capaz de encontrar uma combinação de algoritmos que supera qualquer um dos algoritmos derivados de métodos baseados em decomposição anteriores, selecionando automaticamente a heurística mais adequada para cada função e fase da busca.

Ainda nesse período, [LaTorre, Muelas e Pena \(2012\)](#) propuseram um dos principais métodos para LSGO, o *Multiple Offspring Sampling (MOS)*. Esse método permite a combinação de diferentes metaheurísticas seguindo uma abordagem denominada *High-level Relay Hybrid (HRH)* em que o número de avaliações de cada algoritmo pode ser ajustado dinamicamente. O MOS utiliza um mecanismo, desacoplado do algoritmo principal, para gerar novas soluções candidatas. Isso significa que, dentro de um algoritmo baseado em MOS, vários mecanismos reprodutivos podem ser usados simultaneamente, sendo o algoritmo principal que seleciona entre as técnicas disponíveis a mais apropriada para o problema específico e para a fase da busca. Os pesquisadores verificaram que o MOS apresentava resultados melhores que algoritmos competitivos da época em 12 das 20 funções propostas no *benchmark Special Session on Evolutionary Computation for Large Scale Global Optimization* sendo o campeão do *IEEE CEC Special Session and Competition on Large-Scale Global Optimization* dos anos de 2013 à 2018.

Dentre as diversas metaheurísticas desenvolvidas nos últimos dez anos, algoritmos baseados em Inteligência de Enxame (SI) e Evolução Diferencial (DE) ganharam destaque para problemas LSGO devido sua simplicidade e eficiência ([JIAN; ZHAN; ZHANG, 2020](#)). Como exemplo, [Fister, Žumer et al. \(2012\)](#) desenvolveram um algoritmo memético baseado em colônia de abelhas (MABC) hibridizado com duas heurísticas de busca local: o algoritmo de Nelder-Mead e o passeio aleatório com exploração de direção (RWDE). O MABC foi avaliado na sessão especial para Otimização Global Contínua em Larga Escala do *IEEE Congress on Evolutionary Computation* de 2012. Em seguida, [Korošec e Šilc \(2013\)](#) analisaram o desempenho de um algoritmo de colônia de formigas adaptado para espaços contínuos na Competição de Otimização

de Objetivo Único de Parâmetro Real no *IEEE Congress on Evolutionary Computation* de 2013. [Mohapatra, Das e Roy \(2017\)](#) desenvolveram um otimizador de enxame competitivo modificado (MCSO) com taxa de exploração do espaço de busca melhorada por um critério de competição heurístico. Os resultados empíricos e a análise estatística confirmam o melhor desempenho geral do MCSO em relação a muitas outras metaheurísticas de última geração para problemas LSGO.

[Hanbo Deng et al. \(2019\)](#) propuseram um algoritmo de otimização de enxame de partículas melhorado, denominado Algoritmo de Enxame de Aprendizagem Tendenciosa baseado em Classificação para Otimização em Larga Escala (RBLSO). O RBLSO contém dois tipos de estratégias de aprendizagem: a aprendizagem emparelhada de classificação (RPL) e a aprendizagem centrada enviesada (BCL). No RPL, as piores partículas aprendem ponto a ponto com as melhores partículas de acordo com suas classificações. No BCL, cada partícula aprende com um centro tendencioso que é definido como o centro ponderado de adequação de todo o enxame. Esse operador é utilizado para fortalecer a capacidade exploratória do algoritmo. Os resultados experimentais mostraram que o RBLSO é eficaz na resolução de problemas LSGO. [Jian, Chen et al. \(2021\)](#) propuseram um novo Algoritmo de Enxame de Partícula por Aprendizado Social com esquema de codificação por região adaptativa (SLPSO-ARS). A estratégia de região adaptativa oferece à partícula uma chance maior de descobrir as soluções ótimas próximas e ajuda a controlar a velocidade de convergência de toda a população. [Wang, Wang e Sun \(2022\)](#) propuseram um algoritmo de otimização por enxame de partículas baseado em Aprendizado por Reforço em Níveis (RLLPSO) para problemas LSGO. O RLLPSO usa uma estrutura de população baseada em níveis, onde partículas de níveis inferiores escolhem partículas de níveis superiores como exemplos de aprendizado, aumentando a diversidade da população e a eficiência de busca. Além disso, utiliza o aprendizado por reforço para ajustar automaticamente a estrutura de nível, baseando-se na recompensa do ambiente. Um mecanismo de competição de nível é empregado para otimizar a convergência, permitindo uma seleção preferencial de partículas em níveis mais elitizados como exemplos de aprendizado, acelerando a convergência e reduzindo os custos computacionais. Os resultados experimentais mostram que o RLLPSO supera outros algoritmos avançados de otimização em grande escala em termos de convergência e velocidade. Recentemente, [Li, Wang et al. \(2023\)](#) propuseram, um algoritmo de otimização por enxame de partícula com estratégias de multi-enxame e de sub-enxame adaptativo para gerenciar de forma simultânea e independente a exploração e intensificação de diferentes áreas do espaço de busca. Experimentos abrangentes foram conduzidos com base no *benchmark CEC'2013* e algoritmos evolutivos competitivos de otimização em larga escala. Os resultados demonstram a eficácia dessa estratégia de aprendizagem proposta.

Assim como algoritmos baseados em Inteligência de Enxame (SI), a Evolução Diferencial (DE) provou ser uma abordagem promissora para problemas de otimiza-

ção em Larga Escala com vários aprimoramentos significativos feitos ao longo dos últimos anos (SEGREDO *et al.*, 2018). Alguns trabalhos propunham uma investigação profunda dos efeitos da inicialização das populações no desempenho geral desses algoritmos como forma de melhorar a performance dos mesmos (KAZIMIPOUR; LI; QIN, 2014) (MAUČEC; BREST; BOŠKOVIĆ *et al.*, 2018). Um exemplo interessante dessa abordagem foi proposta por Zhang, Zhan e Zhang (2020). Os pesquisadores desenvolveram uma estratégia de aumento e diminuição gradual da população de acordo com um valor de degradação. Se o valor de degradação de uma solução for grande, significa que a solução tem um valor de aptidão pior ou não melhorou por um longo período de tempo e, portanto, será excluída. Desta forma, o tamanho da população pode ser mantido dentro de um certo intervalo. Outros trabalhos focaram na atualização de operadores conhecidos, como a mutação. Nessa linha de investigação, Hiba, Mahdavi e Rahnamayan (2017) propuseram um esquema de mutação baseado em centro de gravidade. Esse esquema de mutação visa gerar a solução candidata usando o centro de três soluções candidatas selecionadas aleatoriamente, enquanto que Hiba, Ibrahim e Rahnamayan (2019) propuseram cinco diferentes esquemas de mutação baseados em centro dinâmico. Em cada geração, a população é dividida em dois grupos diferentes utilizando estratégias de mutação específicas para cada um. De maneira semelhante, Ma e Bai (2020) propuseram uma evolução diferencial multipopulacional aprimorada com a estratégia de mutação *best-random* (mDE-brM). Nessa abordagem, a população é dividida em três subpopulações com base nos valores de aptidão e também utiliza diferentes estratégias de mutação e parâmetros de controle como forma de melhorar a exploração do espaço de busca multidimensional.

Poucos algoritmos DE obtiveram tanto sucesso em problemas LSGO quanto o SHADE-ILS proposto por Molina, LaTorre e Herrera (2018). Vencedor do *IEEE CEC'2018 Special Session and Competition on Large-Scale Global Optimization*, o SHADE-ILS é um algoritmo híbrido que combina um algoritmo moderno de Evolução Diferencial com um método de busca local escolhido a partir de um conjunto de diferentes métodos de busca. A seleção do método de busca local é dinâmica e leva em consideração a melhoria obtida por cada busca na fase de intensificação anterior para identificar o método mais adequado em cada iteração. Os experimentos apresentaram uma melhoria substancial em relação ao MOS, considerado estado da arte até 2018. Adaptações do SHADE-ILS foram desenvolvidas posteriormente. Por exemplo, Hiba, Rahnamayan *et al.* (2022) propuseram uma mutação baseada em ponto central para o algoritmo SHADE.

Uma linha de investigação que vem ganhando espaço visa tirar proveito de relações de probabilidade no espaço de busca para conduzir a exploração utilizando Algoritmos de Estimção de Distribuição (EDAs). Trabalhos anteriores, como o proposto por Kabán, Bootkrajang e Durrant (2016) e posteriormente aprimorado por Sanyang e Kaban (2019), investigaram o desenvolvimento de um método para produzir algorit-

mos do tipo EDA eficazes e eficientes para problemas de otimização global contínua em larga escala. O conceito é baseado em um conjunto de projeções aleatórias para dimensões mais baixas com o objetivo de diminuir o custo computacional associado à otimização em alta dimensionalidade.

A possibilidade de construção de um espaço de otimização alternativo de custo computacional menor levaram [Dong, Wang e Zhou \(2019\)](#) a propor um EDA baseado no espaço latente (LS-EDA), que transforma o modelo probabilístico multivariado de EDA em seu subespaço latente utilizando a técnica estatística multivariada de Componentes Principais. As dimensões com maior valor projetado contribuem mais para o processo de otimização. O LS-EDA também pode ajudar a reconhecer e compreender a estrutura do problema. Devido à redução de dimensionalidade, seu custo computacional e o tamanho da população podem ser reduzidos efetivamente, enquanto seu desempenho é altamente competitivo em comparação com metaheurísticas de última geração para problemas LSGO.

A ideia de realizar a otimização em um espaço alternativo de menor complexidade permite a investigação de algoritmos que anteriormente não poderiam ser utilizados devido a dimensionalidade do problema, como exemplo pode-se citar os métodos de Agrupamento ou *Clustering*. Na década de 1990, métodos de *Clustering* foram considerados entre as técnicas de otimização global de propósito geral mais eficientes; no entanto, seu uso diminuiu rapidamente nos últimos anos, principalmente devido às dificuldades inerentes de abordagens de agrupamento em espaços de alta dimensionalidade ([BAGATTINI; SCHOEN; TIGLI, 2018](#)). Contudo, [Bagattini, Schoen e Tigli \(2019\)](#) mostraram que métodos de *Clustering* podem ser aplicado em espaços de busca de dimensão reduzida que sejam representativos do espaço original. Com base nessa ideia, [Schoen e Tigli \(2021\)](#) propuseram uma nova abordagem para aplicar métodos de *Clustering* tradicionais para auxiliar uma busca local para variantes de Evolução Diferencial em espaços de busca reduzidos por um método de Redução de Dimensionalidade. O algoritmo resultante, Evolução Diferencial Memética Clusterizada (C-MDE), apresentou resultados melhores que métodos mais conhecidos tanto na qualidade da solução quanto no número de chamadas à função objetivo, contudo apenas 100 dimensões foram consideradas, sendo necessário testes em dimensões maiores.

Os métodos baseados em redução de dimensionalidade podem perder informações de características importantes para a otimização. Um outro fator de relevância encontra-se na dificuldade em garantir que o espaço de busca reduzido retenha as soluções ótimas globais ou soluções de alta qualidade do problema original. Além disso, os algoritmos de otimização geral ainda sofrem com convergência lenta e tendem a cair em ótimos locais. Para resolver os problemas acima, [Feng et al. \(2021\)](#) propuseram um paradigma de busca evolutiva multi-espaço (MSES) para melhorar a eficiência de busca para problemas de otimização em larga escala (LSGO) a partir de um espaço de solução simplificado. No MSES, a direção de busca no espaço do problema original

é guiada em direção a soluções melhores com a assistência proveniente do processo de busca no espaço de solução simplificado. O MSES apresentou melhor desempenho em comparação com outros métodos. No entanto, o desempenho do MSES é apenas satisfatório para resolver certos problemas LSGO. Como uma das razões, sabe-se que o MSES não leva em conta as variadas características estruturais de espaços de busca distintos ao construir o espaço de solução simplificado. Por exemplo, para alguns problemas LSGO com estruturas não-separáveis e complexas, um subespaço de menor dimensionalidade construído por técnicas de redução de dimensionalidade tem mais dificuldade em manter todas as informações estruturais do espaço do problema original à medida que a dimensionalidade diminui. E para alguns problemas de otimização com estruturas simples e separáveis, é mais simples construir um subespaço de menor dimensionalidade que mantenha todas as informações estruturais do problema. Além disso, de acordo com os experimentos realizados com o MSES, variados subespaços de distintas escalas produzem efeitos auxiliares diferentes para a otimização de problemas não-separáveis.

Para remediar as deficiências mencionadas anteriormente, [Song, Wang e Cai \(2022\)](#) propuseram um novo método para LSGO com um subespaço adaptativo à escala baseado em multitarefa evolutiva, denominado EMT-SAS. A multitarefa evolutiva (EMT) é proposta como um novo paradigma de resolução de otimização para estudar como otimizar vários problemas relacionados simultaneamente para melhorar a eficiência e o desempenho da otimização. Isso melhora simultaneamente a taxa de convergência e a qualidade das soluções de busca através do compartilhamento e transferência de conhecimento útil entre problemas relevantes. No método proposto EMT-SAS, os algoritmos evolutivos podem alternar entre subespaços com diferentes escalas de precisão para equilibrar a capacidade de exploração e intensificação do algoritmo. Isso ajuda a explorar sinergias entre tarefas auxiliares de diferentes escalas. Para aproveitar os vários subespaços, a alternância entre diferentes subespaços, foi projetada como uma estratégia de gerenciamento de inter-subespaços, ou seja, a estratégia de subespaço adaptativo à escala (SAS). Os resultados experimentais demonstraram a superioridade do EMT-SAS em comparação com outros algoritmos avançados para LSGO.

Apesar de não utilizar diretamente um espaço reduzido para melhorar a otimização de problemas LSGO, [Rodrigues et al. \(2023\)](#) utilizaram a ideia de aprendizado baseado em oposição (OBL) juntamente com uma metaheurística denominada "água-viva" para melhorar a exploração desses espaços de alta dimensionalidade. A pesquisa possui fins práticos voltados para a etapa de seleção de características em problemas de aprendizado de máquina; problema esse que é comumente de alta dimensionalidade. Os resultados mostraram que a aplicação de OBL pode potencializar a performance de algoritmos desenvolvidos para LSGO. A ideia é promissora e permite uma fácil hibridização com diferentes métodos.

### 2.2.3 Estado da Arte

Esta tese foca em métodos que aprimoram a exploração de espaços de busca de alta dimensionalidade através da criação de novos operadores evolutivos e buscas locais especializadas, ou seja, representados por métodos não baseados em decomposição. Contudo, alguns métodos baseados decomposição foram incluídos no conjunto de teste e comparação devido ao uso de abordagens similares às propostas nesta tese. O Quadro 2 reúne os principais algoritmos competitivos para problema LSGO desenvolvidos nos últimos dez anos e concorrentes da abordagem proposta nesta tese, selecionados através da análise dos trabalhos anteriores e dos algoritmos com boa performance nas principais *benchmarks* para problemas de alta dimensionalidade (*CEC'2013* e *bbob-largescale*). Para critério de classificação, os algoritmos foram diferenciados de acordo com suas semelhanças com o método proposto (LSMDE\*) a partir do uso de: (I) Modelagem de Incerteza; (II) Espaço de busca reduzido; (III) Busca Local; (IV) abordagem não-baseada em Decomposição.

Quadro 2 – Algoritmos concorrentes

Nome	I	II	III	IV	Artigo
CC-CMA-ES	✓		✓		<a href="#">Liu e Tang (2013)</a>
IHDELS			✓	✓	<a href="#">Molina e Herrera (2015)</a>
SHADE-ILS			✓	✓	<a href="#">Molina, LaTorre e Herrera (2018)</a>
C-DEEPSO			✓	✓	<a href="#">Marcelino <i>et al.</i> (2018)</a>
MLSHADE-SPA			✓	✓	<a href="#">Hadi, Mohamed e Jambi (2019)</a>
LS-EDA	✓	✓	✓	✓	<a href="#">Dong, Wang e Zhou (2019)</a>
BICCA		✓			<a href="#">Ge, Zhao <i>et al.</i> (2020)</a>
GL-SHADE			✓	✓	<a href="#">Pacheco-Del-Moral e Coello (2020)</a>
MSES		✓		✓	<a href="#">Feng <i>et al.</i> (2021)</a>
EMT-SAS	✓	✓	✓	✓	<a href="#">Song, Wang e Cai (2022)</a>
LSMDE*	✓	✓	✓	✓	<a href="#">Fonseca <i>et al.</i> (2022)</a>
OJS				✓	<a href="#">Rodrigues <i>et al.</i> (2023)</a>

*I - Modelagem de Incerteza , II - Espaço reduzido, III - Busca Local*

*IV - Não-baseado em Decomposição*

Fonte: Autor

O método proposto parte do conceito inicialmente apresentado por [Bagattini, Schoen e Tigli \(2018\)](#) e [Bagattini, Schoen e Tigli \(2019\)](#) e posteriormente aperfeiçoado por [Schoen e Tigli \(2021\)](#) e [Jian, Chen \*et al.\* \(2021\)](#) acerca da construção de um espaço de busca latente e reduzido dimensionalmente que mantém informações relevantes

do espaço de busca original. Tal característica viabiliza um processo de otimização aproximado no espaço reduzido com um custo computacional menor.

Diferentemente de trabalhos anteriores, esta tese inova ao utilizar uma técnica de decomposição de valores singulares (SVD) para construir de maneira iterativa e adaptativa, um espaço simplificado de baixa dimensionalidade representativo do espaço original, acompanhando o fluxo natural do aprendizado de algoritmos evolutivos.

É importante ressaltar que apesar da técnica de redução se chamar decomposição de valores singulares, o método proposto não é um método baseado em decomposição (ou pelo menos não é no sentido formal definido pelo campo de LSGO). Como mencionado na etapa introdutória da secção 2.2, os métodos baseados em decomposição envolvem a divisão de um problema complexo em subproblemas menores que são resolvidos separadamente. Em contraste, a decomposição por valor singular transforma todo o espaço de problemas, reduzindo a dimensionalidade do problema, mas sem dividi-lo em subproblemas separados. Depois que a transformação é aplicada, o problema reduzido é resolvido como um todo. Logo, a semelhança entre o nome da técnica de redução de dimensionalidade e da subdivisão do campo de LSGO de métodos baseados em decomposição é apenas uma questão de nomenclatura.

Com o objetivo de entender os impactos da otimização em um espaço reduzido, uma análise de escalabilidade foi realizada considerando diferentes cenários de redução e sua influência no resultado final de acordo com diferentes características de espaço de busca.

Além disso, esta tese investiga a exploração do espaço reduzido realizada por um algoritmo de evolução diferencial modificado (DE) auxiliado por uma técnica de Clusterização por Modelo de Misturas Gaussianas (GMM). Como existe perda de informação da solução candidata de baixa dimensionalidade em relação à original, espera-se que o GMM mitigue o problema ao agrupar soluções do espaço reduzido em *clusters* que consideram a incerteza relacionada a proximidade entre estas soluções. Centros de *clusters* promissores armazenam a informação de uma região do espaço de busca de baixa dimensionalidade que possui maiores chances de ter um ótimo local, o que deverá corresponder à uma região também promissora no espaço original se as características estruturais do espaço de busca forem preservadas. Soluções promissoras são re-amostradas para o espaço original e posteriormente exploradas por buscas locais especializadas para otimização em alta dimensionalidade, tais como MTS (TSENG; CHEN, 2008) e L-BFGS-B (MORALES; NOCEDAL, 2011).

## 3 FUNDAMENTAÇÃO TEÓRICA

### 3.1 METAHEURÍSTICAS BASEADAS EM POPULAÇÃO

Metaheurísticas baseadas em população vem ganhando popularidade nos últimos anos para resolver problemas de otimização global em larga escala (OMIDVAR; LI; YAO, 2021a). As metaheurísticas baseadas em população encontram boas soluções iterando e, em seguida, combinando soluções candidatas existentes de um conjunto chamado de população. Os membros mais importantes desta classe são os algoritmos evolutivos (EA). O termo EA abrange uma ampla gama de metaheurísticas que imitam os princípios da evolução natural. Isso inclui algoritmos genéticos (GA) (MITCHELL, 1998), programação genética/evolutiva (GP) (KOZA *et al.*, 1992), Evolução Diferencial (DE) (STORN; PRICE, 1997), e muitos outros.

Formalmente, considere um problema de otimização definido em um espaço de busca  $\chi \subseteq \mathbb{R}^d$  de dimensão  $d$ , onde  $\mathbf{x} \in \chi$  representa um indivíduo viável representado por um vetor de parâmetros  $\mathbf{x} = (x_1, x_2, \dots, x_d)$ . Seja  $f: \chi \rightarrow \mathbb{R}$  a função objetivo que deve ser minimizada (ou maximizada) em relação aos parâmetros de  $\mathbf{x}$ . Nesta tese, as nomenclaturas "solução", "vetor" e "indivíduo" são usadas de maneira intercambiável para representar um vetor no espaço  $d$ -dimensional e possível solução do problema a ser resolvido.

O objetivo é encontrar o vetor de parâmetros  $\mathbf{x}^*$  que otimiza a função objetivo, ou seja,  $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \chi} f(\mathbf{x})$  ou  $\mathbf{x}^* = \arg \max_{\mathbf{x} \in \chi} f(\mathbf{x})$ . Uma metaheurística baseada em população é definida por um processo iterativo que opera sobre uma população de indivíduos, representando soluções candidatas do problema de otimização. A população inicial é composta por  $n$  indivíduos gerados aleatoriamente ou por alguma estratégia heurística.

A cada iteração, os indivíduos da população são avaliados por meio da função objetivo  $f(\mathbf{x})$  e selecionados para reprodução com base em seu desempenho. Os indivíduos selecionados são então submetidos a operadores genéticos, como cruzamento e mutação, que geram novas soluções candidatas.

Formalmente, os operadores genéticos podem ser definidos como segue:

1. **Cruzamento (Recombinação):** Dados dois indivíduos selecionados  $\mathbf{x}_a$  e  $\mathbf{x}_b$ , o operador de cruzamento é denotado por  $c(\mathbf{x}_a, \mathbf{x}_b)$  e produz um ou mais descendentes  $\mathbf{x}_{\text{descendente}}$ , que herdam características dos pais.
2. **Mutação:** Dado um indivíduo selecionado  $\mathbf{x}_i$ , o operador de mutação é denotado por  $m(\mathbf{x}_i)$  e introduz pequenas alterações aleatórias no indivíduo, possibilitando a exploração regiões pouco analisadas do espaço de busca.

Após a aplicação dos operadores genéticos, uma nova população é formada,

substituindo a população anterior. O processo de evolução iterativo prossegue por um número fixo de gerações ou até que um critério de parada seja satisfeito. Os critérios de parada podem incluir um número máximo de iterações, a obtenção de uma solução satisfatória ou a estagnação do processo de evolução.

O objetivo final da metaheurística baseada em população é buscar soluções de alta qualidade, aproximando-se do ótimo global ou oferecendo soluções aceitáveis para o problema de otimização em questão. A diversidade da população, a eficiência na exploração do espaço de busca e a robustez em lidar com funções objetivo complexas são características fundamentais dessas técnicas de otimização.

Para metaheurísticas populacionais, dois fatores muito importantes devem ser levados em consideração para direcionar a busca para áreas promissoras do espaço de soluções na definição de operadores:

- **A exploração**, também chamada de diversificação na literatura, é o processo de guiar a busca para regiões inexploradas. Um algoritmo que realiza exploração insuficiente pode perder regiões importantes, portanto, se o ótimo estivesse em uma dessas regiões, não teria chance de ser encontrado.
- **A intensificação**, é o processo de realizar uma busca minuciosa e intensa por melhores soluções em um ambiente próximo de boas soluções já encontradas. Este processo é essencial para obter maior precisão nas soluções encontradas; porém, quando caem em um ótimo local, tendem a ficar presos nele.

### 3.2 EVOLUÇÃO DIFERENCIAL

Evolução Diferencial (DE), proposta por [Storn e Price \(1997\)](#), é uma metaheurística baseada em população e membro importante da família de algoritmos evolutivos. Essa técnica, fundamentada em populações, realiza a busca pelo ótimo de maneira iterativa. O algoritmo DE é notável por sua facilidade de implementação e eficiência na resolução de uma ampla gama de problemas de otimização, em especial, problemas LSGO ([DE FALCO; DELLA CIOPPA; TRUNFIO, 2019](#); [MOLINA; NESTERENKO; LATORRE, 2019](#); [MAUČEC; BREST, 2019](#)).

Ao contrário de algoritmos evolutivos padrões, o DE elimina a necessidade de codificar cada solução em um cromossomo binário para a realização de operações de cruzamento e mutação, seguidas pela decodificação do cromossomo no vetor de solução. Em vez disso, o DE conduz as operações de cruzamento e mutação diretamente em cada solução, sem codificação e decodificação, tornando-o adequado para resolver problemas de otimização numérica.

Com uma complexidade espacial baixa ([DAS; SUGANTHAN, 2010](#)) e um número reduzido de parâmetros de controle ([KESHK; SINGH; ABBASS, 2018](#); [KRAMER, 2010](#); [BOUKHARI et al., 2018](#)), o algoritmo DE é eficiente e fácil de usar. A técnica emprega

dois parâmetros de controle principais: um fator de escala  $F$ , que governa a mutação, e uma taxa de cruzamento  $CR$ . O parâmetro  $F$  influencia a velocidade de convergência, enquanto o parâmetro  $CR$  tem impacto tanto na convergência quanto na diversidade das populações (ZAHARIE, 2003; KRAMER, 2010).

Em sua execução, o DE utiliza uma população de  $n$  variáveis em vetores  $d$ -dimensionais para cada geração  $G$ .

$$x_{i,G}, i = 1, \dots, n$$

O número de indivíduos  $n$  não muda durante o processo de otimização e a população inicial é escolhida aleatoriamente se não houver informações específicas sobre o problema. Se houver um conjunto de soluções preliminares, essa população será gerada de acordo com um desvio padrão aleatório dos dados fornecidos pelas soluções preliminares.

De maneira geral, Evolução Diferencial é um esquema para geração de vetores paramétricos experimentais por meio da diferença ponderada de dois vetores pertencentes à população com um terceiro membro da mesma população. Posteriormente, se for identificado melhoria ao avaliar o vetor resultante do processo anterior, então o vetor original da população com a qual a comparação está sendo feita é substituído por este novo vetor. A Evolução Diferencial implementa três operações fundamentais: mutação, cruzamento e seleção, que são discutidos a seguir.

### 3.2.1 Mutação

Para cada vetor  $x_{i,G}$ , gera-se um vetor mutante de acordo com a seguinte regra:

$$v_{i,G+1} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}) \quad (1)$$

Onde os índices  $r1, r2, r3 \in 1, \dots, n$  são gerados aleatoriamente e  $F > 0$ . O fator  $F$  é uma constante real definida no intervalo  $[0, 2]$  por meio da qual se controla a expansão da variação diferencial ( $x_{r2} - x_{r3}$ ). Essa estratégia de geração é conhecida como *DE/rand/1* (QIN; SUGANTHAN, 2005).

A fim de garantir uma maior diversidade das soluções contidas no espaço de busca e promover um melhor equilíbrio entre exploração e intensificação, surgiram novas estratégias para a aplicação do processo de mutação descrito na Equação 1, entre as quais se destacam: (QIN; SUGANTHAN, 2005).

- *DE/best/1* definida mediante a equação  $v_{i,G} = x_{best,G} + F \cdot (x_{r2,G} - x_{r3,G})$
- *DE/current/1* definida mediante a equação  $v_{i,G} = x_{i,G} + F \cdot (x_{best,G} - x_{i,G}) + F \cdot (x_{r1,G} - x_{r2,G})$
- *DE/rand/2* definida mediante a equação  $v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G} + x_{r4,G} - x_{r5,G})$

- *DE/best/2* definida mediante a equação  $v_{i,G} = x_{best,G} + F \cdot (x_{r2,G} - x_{r3,G} + x_{r4,G} - x_{r5,G})$

O DE utilizado no LSMDE utiliza um novo tipo de mutação denominado *DE/promising/2* a ser detalhado no capítulo 5 de proposta de tese.

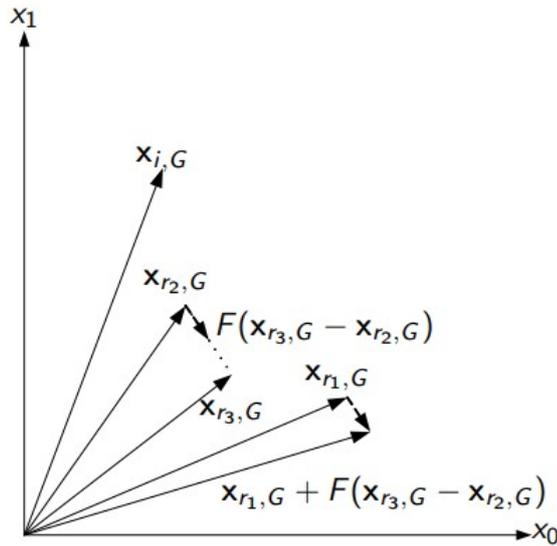


Figura 2 – Ilustração do processo de mutação

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

### 3.2.2 Cruzamento

A fim de manter a diversidade, o cruzamento dos vetores é introduzido para o vetor de teste resultante  $u$  tal que:

$$u_{i,G+1} = (u_{i1,G+1}, u_{i2,G+1}, \dots, u_{id,G+1})$$

em que  $j = 1, \dots, d$  e:

$$u_{ij,G+1} = \begin{cases} v_{ij,G+1} & \text{se } r_j \leq CR \text{ ou } j = I_i \\ x_{ij,G} & \text{se } r_j > CR \text{ ou } j \neq I_i \end{cases} \quad (2)$$

onde  $j = 1, \dots, d$ ,  $r_j \sim U(0, 1)$ ,  $CR \in [0, 1]$  é uma constante definida pelo usuário e  $I_i$  é um índice aleatoriamente escolhido  $\in 1.., d$ , o que garante que  $u_{i,G+1}$  recebe pelo menos uma componente de  $v_{i,G+1}$ . Seja  $x_{i,G}$  o vetor-alvo sob análise e  $v_{i,G+1}$  o respectivo vetor mutado obtido a partir da Equação 1, A Figura 3 ilustra o processo de geração do vetor teste  $u_{i,G+1}$ .

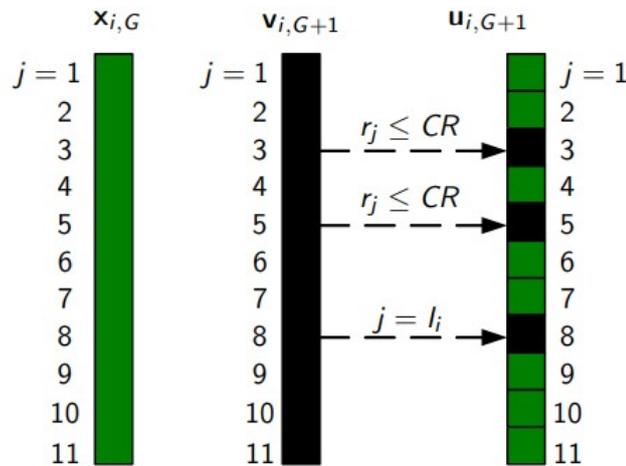


Figura 3 – Exemplo de um processo de cruzamento

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

### 3.2.3 Seleção

Após as etapas de mutação e cruzamento, nas quais todos os  $n$  vetores serviram como vetor-alvo, a seleção dos vetores que serão preservados para a próxima geração  $G + 1$  é feita usando um critério guloso definido abaixo:

$$x_{i,G+1} = \begin{cases} u_{i,G+1}, & \text{se } f(u_{i,G+1}) < f(x_{i,G}) \\ x_{i,G}, & \text{caso contrário} \end{cases} \quad (3)$$

onde  $u_{i,G+1}$  é o vetor teste obtido através da Equação 2,  $x_{i,G}$  é o vetor-alvo  $i$  que fará parte da geração  $G + 1$  e  $f$  é uma função objetivo utilizada para avaliar as soluções candidatas.

## 3.3 APRENDIZADO BASEADO EM OPOSIÇÃO

Metaheurísticas populacionais e baseadas em uma solução única tentam resolver um problema de otimização para encontrar uma ou mais soluções ótimas desconhecidas. Como não há conhecimento a priori sobre a localização da solução ou a paisagem do problema, o algoritmo começa a resolver o problema usando uma única solução ou população de soluções candidatas. Em metaheurísticas baseadas em solução única, a solução candidata é um único ponto no espaço de busca, o algoritmo guia a solução candidata através do espaço de busca para obter a solução ótima. Na prática, a solução candidata é um ponto aleatório uniforme que é gerado em algum lugar no espaço do problema. Existem outros métodos para gerar a solução candidata inicial, contudo, inicialização aleatória tem sido amplamente utilizada por um longo período de tempo.

O aprendizado baseado em oposição (*opposition-based learning-OBL*) foi proposto por [Hamid R Tizhoosh \(2005a\)](#) para melhorar metaheurísticas considerando as

soluções candidatas atuais e seus opostos correspondentes simultaneamente durante a inicialização e os processos evolutivos. De um modo geral, o esquema OBL considera a estimativa atual e sua estimativa oposta correspondente para convergir para uma solução ótima para uma determinada função objetivo. Muitos algoritmos de aprendizado de máquina foram aprimorados utilizando esse conceito, tais como: Aprendizado por Reforço (RL) (TIZHOOSH, H. R., 2005b), Redes Neurais Artificiais (ANN) (VENTRESCA; TIZHOOSH, 2006), sistemas fuzzy (TIZHOOSH, H. R., 2009) e variantes de métodos de otimização conhecidos, tais como: Algoritmos Genéticos (GA) (PATEL; RAGHUWANSHI; MALIK, 2012), Evolução Diferencial (DE) (DENG, W. *et al.*, 2022), Otimização por Enxame de Partículas (PSO) (JABEEN; JALIL; BAIG, 2009), Colônia de Formigas (ACS) (ZHAO *et al.*, 2022) e Recozimento simulado (SA) (VENTRESCA; TIZHOOSH, 2007).

O oposto da solução candidata, ou candidato-oposto, é o ponto espelho da solução candidata a partir do centro do espaço de busca. A Figura 4 apresenta um intervalo de busca de  $[a, b]$ , a solução candidata,  $x$ , e o candidato-oposto  $\hat{x}$ . Matematicamente, o candidato-oposto pode ser definido da seguinte forma em um espaço de busca unidimensional (TIZHOOSH, H., 2005):

$$\hat{x} = a + b - x \quad (4)$$

A definição pode ser estendida para o espaço  $d$ -dimensional da seguinte forma (TIZHOOSH, H., 2005):

$$\hat{x}_j = a_j + b_j - x_j \quad (5)$$

onde  $x_j \in [a_j, b_j]$  e  $j = 1, \dots, d$  e  $d$  é a dimensão do problema.

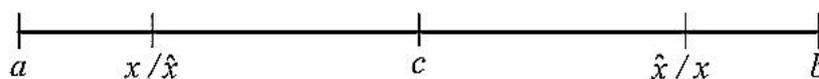


Figura 4 – Ilustração visual (em 1D) do espaço de busca  $[a, b]$ . O  $c$  é o centro do intervalo,  $x$  é a solução candidata e  $\hat{x}$  é o candidato-oposto.

Fonte: Adaptado de Mahdavi, Rahnamayan e Deb (2018)

### 3.3.1 Segmentações do espaço de busca baseados em oposição

Nesta subseção, serão explicados alguns esquemas de segmentação de espaços de busca baseados em oposição que foram utilizados na construção do método LSMDE proposto nesta tese.

### 3.3.1.1 Ponto central

O Ponto central é um ponto único, localizado no centro do espaço de busca, mostrado por  $c$  na Fig. 4.1. O ponto único é definido no espaço de busca unidimensional da seguinte forma:

$$c = \frac{(a + b)}{2} \quad (6)$$

e pode ser estendido para o espaço  $d$ -dimensional da seguinte forma:

$$c_j = \frac{(a_j + b_j)}{2} \quad (7)$$

onde  $j = 1, \dots, d$  e  $d$  é a dimensão do espaço de busca.

É importante notar que o cálculo do ponto central é independente a partir do candidato  $x$  ou do candidato-oposto  $\hat{x}$ .

### 3.3.1.2 Intervalo baseado no centro

O intervalo baseado no centro (1D) (ou região, no caso 2D) é definido como o intervalo/região entre o candidato,  $x$ , e o candidato-oposto,  $\hat{x}$  ( $[x, \hat{x}]$ ), conforme mostrado nas Figuras 5a e 5b para 1D e 2D, respectivamente. A solução candidata nesse caso, denotada por  $x^{cb}$ , é um ponto aleatório uniforme gerado dentro do intervalo/região baseado no centro  $[x, \hat{x}]$ .

A formulação matemática de  $x^{cb}$  no espaço de busca unidimensional é mostrada da seguinte forma:

$$x^{cb} = \begin{cases} x + (\hat{x} - x) \times \text{rand}(0, 1) & \text{se } x \leq \hat{x} \\ x + (x - \hat{x}) \times \text{rand}(0, 1) & \text{caso contrário} \end{cases} \quad (8)$$

Pode-se estender essa definição para o espaço  $d$ -dimensional da seguinte forma:

$$x_j^{cb} = \begin{cases} x_j + (\hat{x}_j - x_j) \times \text{rand}(0, 1) & \text{se } x_j \leq \hat{x}_j \\ x_j + (x_j - \hat{x}_j) \times \text{rand}(0, 1) & \text{caso contrário} \end{cases} \quad (9)$$

onde  $x_j \in [a_j, b_j]$  e  $j = 1, \dots, d$  e  $d$  é a dimensão do espaço de busca.

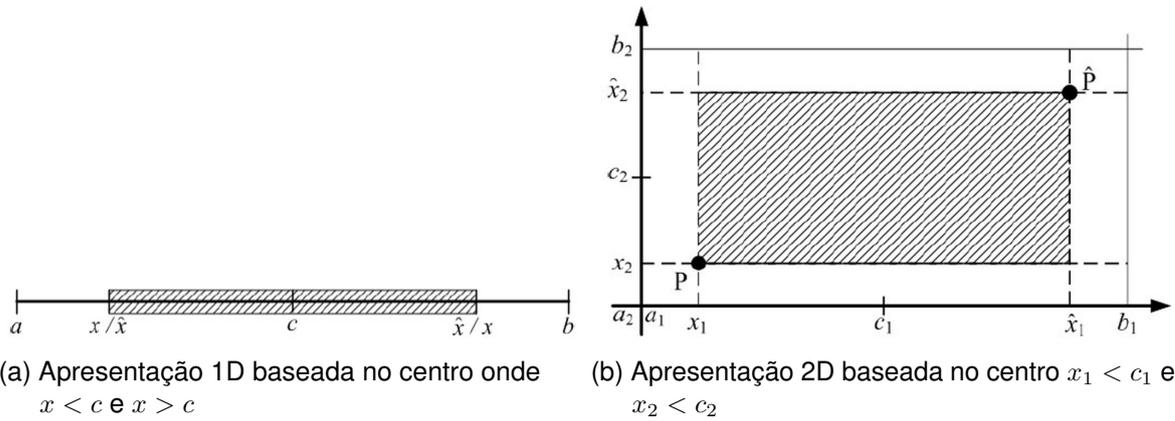


Figura 5 – A apresentação visual em 1D (a) e 2D (c) do esquema de segmentação baseado no centro, onde  $x_1 \in [a_1, b_1]$  e  $x_2 \in [a_2, b_2]$ .

Fonte: Adaptado de [Mahdavi, Rahnamayan e Deb \(2018\)](#)

### 3.3.1.3 Intervalo oposto modular

O intervalo/região oposto modular é definido como o intervalo entre a solução candidata,  $x$ , mais ou menos a metade do intervalo do espaço de busca  $[a, b]$ , conforme apresentado pelas Figuras 6a, 6b e 6c. Em outras palavras, o intervalo oposto modular abrange metade do espaço de busca. O candidato-oposto modular, denotado  $x^{mo}$ , é um ponto aleatório uniformemente gerado dentro do intervalo oposto modular  $[x, (x+c-a)]$ , onde  $c$  indica o centro do espaço de busca.

O cálculo matemático de  $x^{mo}$  no espaço de busca unidimensional é mostrado a seguir:

$$x^{mo} = \begin{cases} x + (c - a) \times \text{rand}(0,1) & \text{se } x \leq c \\ (x - c + a) + (c - a) \times \text{rand}(0,1) & \text{caso contrário} \end{cases} \quad (10)$$

e pode ser estendido para o espaço  $d$ -dimensional da seguinte forma:

$$x_j^{mo} = \begin{cases} x_j + (c_j - a_j) \times \text{rand}(0,1) & \text{se } x_j \leq c_j \\ (x_j - c_j + a_j) + (c_j - a_j) \times \text{rand}(0,1) & \text{caso contrário} \end{cases} \quad (11)$$

onde  $c_j = \frac{(a_j+b_j)}{2}$ ,  $x_j \in [a_j, b_j]$  e  $j = 1, \dots, d$  e  $d$  é a dimensão do espaço de busca.

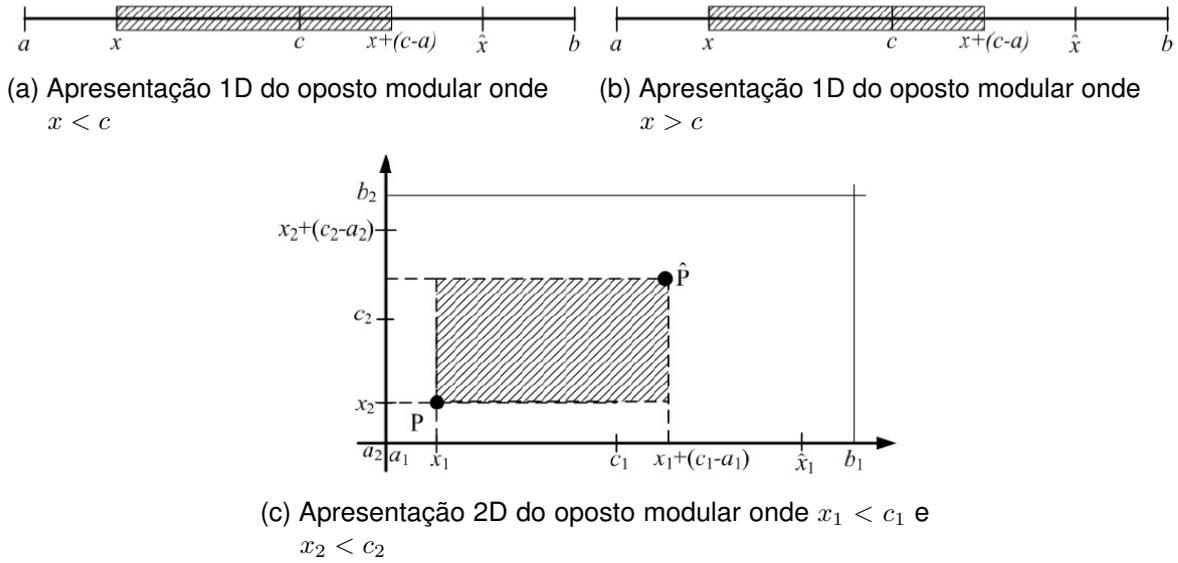


Figura 6 – Ilustração em 1D ((a) e (b)) e 2D (c) da região de busca oposta modular, onde  $x_1 \in [a_1, b_1]$  e  $x_2 \in [a_2, b_2]$ .

Fonte: Adaptado de [Mahdavi, Rahnamayan e Deb \(2018\)](#)

### 3.3.1.4 Intervalo quase oposto

O intervalo/região quase oposto é definido como o intervalo/região entre o centro  $c$  e o candidato-oposto,  $\hat{x}$ , conforme representado nas Figuras 7a, 7b e 7c, para 1D e 2D, respectivamente. A solução candidata quase-oposta,  $x^{qo}$ , é um ponto aleatório uniforme gerado dentro do intervalo/região de quase-opostos ( $[c, \hat{x}]$  onde  $c < \hat{x}$ ). O cálculo matemático de  $x^{qo}$  no espaço de busca unidimensional é mostrado a seguir:

$$x^{qo} = \begin{cases} c + (\hat{x} - c) \times rand(0,1) & \text{se } x \leq c \\ \hat{x} + (c - \hat{x}) \times rand(0,1) & \text{caso contrário} \end{cases} \quad (12)$$

E pode ser estendido para o espaço de busca  $d$ -dimensional da seguinte forma:

$$x_j^{qo} = \begin{cases} c_j + (\hat{x}_j - c_j) \times rand(0,1) & \text{se } x_j \leq c_j \\ \hat{x}_j + (c_j - \hat{x}_j) \times rand(0,1) & \text{caso contrário} \end{cases} \quad (13)$$

onde  $c_j = \frac{(a_j+b_j)}{2}$ ,  $\hat{x}_j \in [a_j, b_j]$  e  $j = 1, \dots, d$  e  $d$  é a dimensão do espaço de busca.

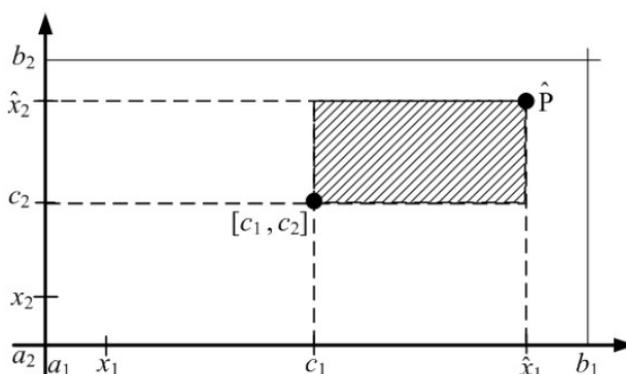
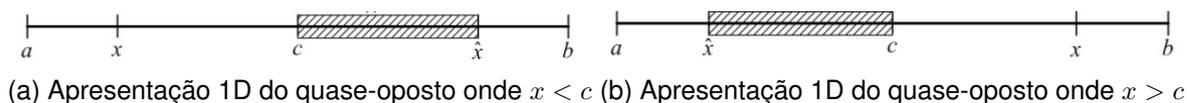


Figura 7 – Ilustração em 1D ((a) e (b)) e 2D (c) da região de busca quase-oposta, onde  $x_1 \in [a_1, b_1]$  e  $x_2 \in [a_2, b_2]$ .

Fonte: Adaptado de [Mahdavi, Rahnamayan e Deb \(2018\)](#)

Em suma, ao considerar não apenas a solução candidata, mas também seu oposto, cria-se uma estratégia bidirecional de busca que amplia a capacidade do algoritmo de escapar de ótimos locais e convergir de maneira mais eficaz para um ótimo global. Essa abordagem dual permite uma representação mais rica e variada do espaço de soluções, tornando possível a identificação de regiões do espaço de busca que poderiam permanecer inexploradas em métodos tradicionais de otimização. Portanto, o Aprendizado por Oposição oferece um mecanismo robusto para aprimorar a diversificação de soluções, o que é crucial para o sucesso em problemas de otimização em larga escala, onde a complexidade e a alta dimensionalidade tornam a exploração eficiente do espaço de busca um desafio significativo.

### 3.4 REDUÇÃO DE DIMENSIONALIDADE BASEADO EM DECOMPOSIÇÃO DE VALOR SINGULAR

A redução de dimensionalidade é o processo de mapear um conjunto de dados de alta dimensão para um espaço de dimensão inferior, preservando grande parte da estrutura importante. Em Estatística e Aprendizado de Máquina, isso geralmente se refere ao processo de encontrar algumas direções nas quais um vetor aleatório de alta dimensão tem variância máxima.

Suponha um conjunto de dados representado como linhas de uma matriz  $A : a_1, \dots, a_n$  com dimensões  $n \times d$  e  $n > d$ . O objetivo é encontrar o subespaço  $k$ -dimensional tal que a soma do quadrado da distância entre os dados e o subespaço é minimizada. Supondo-se que o subespaço  $k$ -dimensional seja gerado por  $k$

vetores  $v_1, v_2, \dots, v_k$ , tem-se por objetivo encontrar  $v_1, \dots, v_k$  e  $(n \cdot k)$  coeficientes  $b_{ij}$  que minimizem  $\sum_{i=1}^n \left\| a_i - \sum_{j=1}^k b_{ij} v_j \right\|^2$  (LANGE; LANGE, 2010).

**Teorema 3.1.** *Considere o vetor  $v$  e o subespaço  $H$ . Dado que  $dist$  é a distância entre  $v$  e o subespaço e  $proj$  é o comprimento da projeção de  $v$  em  $H$ , tem-se que*

$$\|v\|^2 = dist^2 + proj^2$$

O teorema promove uma visão alternativa do subespaço de melhor ajuste. Uma vez que os comprimentos dos vetores são fixos, minimizando a soma do quadrado das distâncias entre os pontos e os subespaços é equivalente a maximizar a soma dos quadrados dos comprimentos das projeções.

### 3.4.1 Vetor singular

Considerando o caso em que  $k = 1$ , ou seja, procura-se uma linha unidimensional indo através da origem tal que a soma dos quadrados das distâncias entre os pontos e a reta é minimizado. Suponha que  $v$  é um vetor ao longo dessa reta. O comprimento da projeção de  $a_i$  na reta é  $|\langle a_i, v \rangle|$ . Como definido anteriormente, procura-se por  $v$  que maximize  $\sum_{i=1}^n |\langle a_i, v \rangle|^2 = \|Av\|^2$ . O primeiro vetor singular de  $A$  é definido como

$$v_1 = \arg \max_{\|v\|=1} \|Av\|$$

O maximizador não é único. Para qualquer vetor  $v$ , o vetor  $-v$  possui o mesmo valor objetivo. Logo, pode-se selecionar um maximizador arbitrário e referi-lo como primeiro vetor singular. O valor  $\varrho_1 = \|Av_1\|$  é chamado de primeiro vetor singular de  $A$ .

Pode-se definir os outros vetores singulares recursivamente. O Segundo vetor singular é a reta de melhor ajuste entre as ortogonais a  $v_1$ :

$$v_2 = \arg \max_{\|v\|=1, v \perp v_1} \|Av\|, \varrho_2 = \|Av_2\|$$

e assim por diante:

$$v_i = \arg \max_{\|v\|=1, v \perp v_1, \dots, v_{i-1}} \|Av\|$$

Observa-se que para qualquer  $k$ , o subespaço  $k$ -dimensional de melhor ajuste é gerado pelos primeiros  $k$  vetores singulares.

**Teorema 3.2.** *A extensão dos primeiros  $k$  vetores singulares é um subespaço dimensional  $k$  de melhor ajuste.*

Pode-se provar o teorema usando indução. Para  $k = 1$ , o teorema vale por definição. Em seguida, assume-se que o teorema é verdadeiro para  $k = m$  e prova-se o teorema para  $k = m + 1$ . Suponha que  $W$  seja um subespaço dimensional  $m + 1$  de melhor ajuste, pode-se escolher uma base para  $W$  como segue: Primeiro, seja

$w_{m+1}$  um vetor unitário em  $W$  ortogonal às projeções de  $v_1, \dots, v_m$  sobre  $W$ . Em seguida, continua-se escolhendo vetores em  $W$  que são ortogonais aos vetores anteriores para obter uma base ortogonal completa de  $W$  :  $w_1; w_2, \dots, w_{m+1}$ .

$w_{m+1}$  é ortogonal a  $v_1, \dots, v_m$  porque cada  $v_i$  é a soma de dois componentes: sua projeção em  $W$  e seu componente que é ortogonal a  $W$  e  $w_{m+1}$  que é ortogonal a ambos.

Por hipótese, a extensão  $v_1, \dots, v_m$  é o subespaço  $d$ -dimensional de melhor ajuste então:

$$\sum_{i=1}^m \|Av_i\|^2 \geq \sum_{i=1}^m \|Aw_i\|^2$$

Uma vez que  $w_{m+1}$  é ortogonal a  $v_1, \dots, v_d$  e  $v_{m+1}$  é a melhor reta de ajuste entre os vetores ortogonais a  $v_1, \dots, v_m$ :

$$\|Av_{m+1}\|^2 \geq \|Aw_{m+1}\|^2$$

Assim:

$$\sum_{i=1}^{m+1} \|Av_i\|^2 \geq \sum_{i=1}^{m+1} \|Aw_i\|^2$$

Uma vez que  $W$  é um melhor subespaço ajustado, a extensão  $v_1, \dots, v_{m+1}$  é também um melhor subespaço ajustado.

Os vetores  $v_1, \dots, v_d$  são chamados de vetores singulares à direita de  $A$ . Os vetores  $Av_i$  são também uteis e normaliza-se para o comprimento igual a 1:

$$u_i = \frac{1}{\sigma_i} Av_i$$

Os  $u_i$  são chamados vetores singulares à esquerda de  $A$ . Pode-se mostrar que  $u_i$  são ortogonais e  $u_i$  é o vetor unitário que maximiza  $\|A^T u\|$  entre vetores ortogonais a  $u_1, \dots, u_{i-1}$ . Esses vetores formam a decomposição de  $A$  em matrizes de posto 1 (LANGE; LANGE, 2010):

$$A = \sum_i \sigma_i u_i v_i^T$$

**Matrizes de posto 1:** é precisamente uma matriz diferente de zero em que todas as linhas e colunas são (não necessariamente integrais) múltiplas umas das outras. Uma definição equivalente de uma matriz  $m \times n$  de posto 1 é como o produto externo (ou produto vetorial)  $uv^T$  de um  $m$ -vetor  $u$  e um  $n$ -vetor  $v$ , como ilustrado na Equação 14:

$$A = uv^T \begin{bmatrix} \dots & u_1 v^T & \dots \\ \dots & u_2 v^T & \dots \\ & \vdots & \\ \dots & u_m v^T & \dots \end{bmatrix} = \begin{bmatrix} \dots & \dots & \dots \\ v_1 u & v_2 u & \dots & v_n u \\ \dots & \dots & & \dots \end{bmatrix} \quad (14)$$

**Matrizes de posto  $k$ :** Uma matriz  $A$  tem posto  $k$  se puder ser escrita como a soma de  $k$  matrizes de posto 1, e não puder ser escrita como a soma de  $k \leq 1$  ou menos matrizes de posto 1.

Reformulado em termos de multiplicação de matrizes, uma definição equivalente é que  $A$  pode ser escrita como (ou fatorada em) o produto de uma longa e estreita ( $m \times k$ ) matriz  $Y$  e uma curta e longa ( $k \times n$ ) matriz  $Z^T$  (e que  $A$  não pode ser igualmente fatorada no produto de  $m \times (k \leq 1)$  e  $(k \leq 1) \times n$  matrizes).

Uma aproximação de baixo posto fornece uma versão comprimida da matriz (com perda). A matriz original  $A$  é descrita por  $(m \cdot n)$  elementos, enquanto a descrição de  $Y$  e  $Z^T$  requer apenas  $k(m + n)$  elementos. Quando  $k$  é pequeno em relação a  $m$  e  $n$ , substituir o produto de  $m$  e  $n$  pela sua soma é uma grande vantagem. Por exemplo, quando  $A$  representa um espaço de busca com alta dimensionalidade,  $m$  e  $n$  podem ser da ordem de dezenas de milhares ou mais. Um valor menor de  $k$  geralmente é suficiente para obter aproximações que se pareçam muito com o espaço original.

### 3.4.2 Decomposição em Valores Singulares (SVD)

A decomposição em valores singulares (SVD) é uma importante ferramenta matemática na Ciência da Computação. É um método para transformar dados correlacionados em um conjunto de dados não correlacionados. A SVD é definida como se segue para uma matriz  $A \in \mathbb{R}^{m \times n}$ :

$$A = USV^T \quad (15)$$

onde  $U \in \mathbb{R}^{m \times m}$  e  $V \in \mathbb{R}^{n \times n}$  são matrizes ortogonais (i.e.,  $U^T U = I_m$  e  $V^T V = I_n$ ), e  $S \in \mathbb{R}^{m \times n}$  é uma matriz diagonal de valores singulares. Os valores singulares na matriz  $S$  são ordenados em ordem decrescente, e são as raízes quadradas dos autovalores de  $A^T A$  (LANGE; LANGE, 2010).

As colunas de  $U$  são os vetores singulares à esquerda de  $A$ . As colunas de  $V$  (ou seja, as linhas de  $V^T$ ) são os vetores singulares à direita de  $A$ . As entradas de  $S$  são os valores singulares de  $A$ . Assim, com cada vetor singular (à esquerda ou direita) existe um valor singular associado. O primeiro vetor singular refere-se a aquele associado ao maior valor singular e assim por diante. O SVD expressa  $A$  tal que a fatorização  $A = USV^T$  é equivalente a expressão:

$$A = \sum_{i=1}^{\min(m,n)} s_i \cdot u_i \cdot v_i^T$$

onde  $s_i$  é o  $i$ -ésimo valor singular e  $u_i$ ;  $v_i$  são os vetores singulares à esquerda e direita correspondentes. Ou seja, o SVD expressa  $A$  como uma combinação linear não negativa de  $\min m, n$  matrizes de posto 1, com os valores singulares fornecendo os multiplicadores e os produtos externos dos vetores singulares fornecendo as matrizes de posto 1.

Dado que SVD expressa uma matriz  $A$  como uma soma de matrizes de posto 1 (ponderadas pelos valores singulares correspondentes), pode-se manter apenas os primeiros  $k$  termos mais significativos que representam  $A$ , ou seja, realizar uma aproximação de baixo posto:

$$\tilde{A} = \sum_{i=1}^k s_i \cdot u_i \cdot v_i^T$$

onde, assumi-se que os valores singulares foram ordenados ( $s_1 \geq s_2 \geq \dots s_{\min(m,n)} \geq 0$ ), e  $u_i$  e  $v_i$  denotam os  $i$ -ésimos vetores singulares à esquerda e direita. Como a soma de  $k$  matrizes de posto 1,  $\tilde{A}$  tem também posto  $k$ .

Armazenar as matrizes  $U_k S_k V_k^T$  requer complexidade espacial de  $O(k(m+n))$ , em contraste com  $O(m \cdot n)$  requerido para armazenar a matriz original  $A$ . Isto é um grande ganho quando  $k$  é relativamente pequeno e o produto  $(m \cdot n)$  é relativamente grande. A definição do valor de  $k$  é uma tarefa importante pois ela está diretamente relacionada com a margem de erro aceitável entre a matriz original e a matriz de baixo posto  $k$ . Para definir o melhor valor  $k$ , pode-se usar o lema de Johnson-Lindenstrauss.

### 3.4.3 Lema de Johnson-Lindenstrauss

Uma parte importante da compreensão da redução de dimensionalidade é o lema de Johnson–Lindenstrauss (JL). O JL afirma que quaisquer  $n$  pontos no espaço euclidiano de alta dimensão podem ser mapeados em  $k$  dimensões onde  $k \geq O(\log n / \epsilon^2)$  sem distorcer a distância entre quaisquer dois pontos mais do que um fator de  $1 \pm \epsilon$  (DASGUPTA; GUPTA, 2003).

**Lema 3.3** (Johnson-Lindenstrauss). *Para qualquer  $0 < \epsilon < 1$  e inteiro  $n$ , seja  $k$  um inteiro positivo tal que  $k \geq 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \log n$ . Então, para qualquer conjunto  $A$  de  $n$  pontos em  $\mathbb{R}^d$ , existe uma função  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  tal que para todos os  $\vec{x}, \vec{y} \in A$ ,*

$$(1 - \epsilon) |\vec{x} - \vec{y}|^2 \leq |f(\vec{x}) - f(\vec{y})|^2 \leq (1 + \epsilon) |\vec{x} - \vec{y}|^2.$$

*Além disso, esta função pode ser encontrada em tempo polinomial aleatório.*

Prova: Construa uma projeção aleatória sobre  $k$  subespaços dimensionais. Prove que o valor esperado da distância euclidiana da projeção aleatória é igual à distância

euclidiana do subespaço original. Prove que a variância da distância euclidiana é maior que o fator de erro especificado apenas com uma probabilidade  $F = 2/n^2$  tal que o limite de união desta probabilidade em todos os pares de pontos é menor que  $1 \leq 1/n$ .

O lema de JL afirma que, depois de fixar um nível de erro  $\epsilon$ , pode-se mapear uma coleção de pontos de um espaço (não importa quão alta seja a dimensão  $d$ ) para um espaço dimensionalmente menor, alterando apenas a distância entre quaisquer dois pontos por um fator de  $1 \pm \epsilon$ .

Algo a ser observado sobre o lema JL é que ele é independente dos próprios vetores  $x$ ; o lema de JL depende apenas da dimensão  $d$ , número de vetores  $n$  e distorção  $\epsilon$ .

### 3.5 MODELOS DE MISTURA

A redução de dimensionalidade é uma técnica amplamente utilizada para simplificar conjuntos de dados de alta dimensão, tornando-os mais tratáveis e visualmente interpretáveis. No entanto, ao transformar dados para um espaço de menor dimensão, existe o risco inerente de perder informações cruciais. Modelos de Misturas, com sua abordagem probabilística, oferecem um meio de mitigar os efeitos dessa perda, permitindo uma interpretação dos dados mesmo em espaços de dimensão reduzida.

Segundo [McLachlan, Lee e Rathnayake \(2019\)](#), dado um conjunto de dados  $X = x_1, \dots, x_n$  em que  $n$  é o número de observações de uma variável aleatória  $d$ -dimensional  $x$ . A variável aleatória  $x_n$  é assumida ser distribuída de acordo com a mistura de  $C$  componentes. Cada componente (*cluster*) é matematicamente representado por uma distribuição paramétrica. Uma distribuição individual usada para modelar um *cluster* específico é geralmente chamada como uma distribuição de componentes. Todo o conjunto de dados é, portanto, modelado por uma mistura dessas distribuições. Formalmente, a distribuição de mistura, ou função de densidade de probabilidade de  $x_n$  pode ser escrita como:

$$p(x_n) = \sum_{c=1}^C \pi_c p(x_n | \theta_c) \quad (16)$$

onde  $\pi_1, \dots, \pi_C$  são as probabilidades de mistura (também chamados de coeficientes de mistura ou pesos), cada  $\theta_c$  é um conjunto de parâmetros especificando o  $c$ -ésimo componente e  $p(x_n | \theta_c)$  é a distribuição do componente. Para serem probabilidades válidas, as probabilidades de mistura  $\pi_c$  devem satisfazer

$$0 \leq \pi_c \leq 1 \quad (c = 1, \dots, C) \text{ e } \sum_{c=1}^C \pi_c = 1$$

Uma maneira de gerar uma amostra aleatória  $x_n$  com o modelo de mistura dado pela Equação 16 pode ser feita da seguinte forma: Seja  $z_n$  uma variável aleatória

categórica assumindo os valores  $1, \dots, C$  com probabilidades  $p(z_n = c) = \pi_c$  (também denotado como  $p(z_{nc} = 1) = \pi_c$ , respectivamente). Suponha que a distribuição condicional de  $x_n$  dado  $z_n = c$  seja  $p(x_n|\theta_c)$ . Então a distribuição marginal de  $x_n$  é obtida somando a distribuição conjunta sobre todos os valores possíveis de  $z_n$  para dado  $p(x_n)$  como na Equação 16. Nesse contexto, a variável  $z_n$  pode ser considerada como o componente (ou *cluster*) da amostra aleatória  $x_n$ . Em vez de usar uma única variável categórica  $z_n$ , introduz-se um vetor aleatório binário  $C$ -dimensional  $z_n$  para denotar o componente para  $x_n$ . A variável aleatória  $C$ -dimensional  $z_n$  tem uma representação de 1 a  $C$ , na qual um dos elementos  $z_{nc} = (z_n)_c$  é igual a 1, e todos os outros elementos iguais a 0, denotando que o componente de origem de  $x_n$  é igual a  $c$ . Por exemplo, dado uma variável com  $C = 5$  *clusters* e uma observação particular  $x_n$  da variável corresponder ao *cluster* onde  $z_{n4} = 1$ , então  $z_n$  será representado por  $z_n = (0, 0, 0, 1, 0)^T$ . Observa-se que os valores de  $z_{nc}$  satisfazem  $z_{nc} \in \{0, 1\}$  e  $\sum_c z_{nc} = 1$ . Como  $z_n$  usa uma representação de 1 a  $C$ , a distribuição marginal sobre  $z_n$  é especificada em termos de probabilidades de mistura  $\pi_c$ , tais que

$$p(z_n) = \pi_1^{z_{n1}} \pi_2^{z_{n2}} \dots \pi_C^{z_{nC}} = \prod_{c=1}^C \pi_c^{z_{nc}} \quad (17)$$

Similarmente, a distribuição condicional de  $x_n$  dado  $z_n$  pode ser escrita na forma:

$$p(x_n|z_n) = \prod_{c=1}^C p(x_n|\theta_c)^{z_{nc}} \quad (18)$$

A distribuição conjunta é dada por  $p(z_n)p(x_n|z_n)$  e a distribuição marginal de  $x_n$  é obtida por:

$$p(x_n) = \sum_{z_n} p(z_n)p(x_n|z_n) = \sum_{c=1}^C \pi_c p(x_n|\theta_c). \quad (19)$$

Assim, a distribuição marginal de  $x$  é uma formulação equivalente do modelo de mistura envolvendo uma variável latente explícita. Do ponto de vista do processo generativo, um determinado conjunto de pontos de dados poderia ter sido gerado repetindo o seguinte procedimento  $N$  vezes, uma vez para cada ponto de dados  $x_n$ :

- Escolher um componente oculto (isto é, *cluster*)  $z_n \sim Mult_C(1, \pi)$ . Isso seleciona o  $c$ -ésimo componente a partir do qual se observa o ponto  $x_n$ .
- Amostram um ponto de dados  $x_n$  do  $c$ -ésimo componente de acordo com a distribuição condicional  $p(x_n|\theta_c)$

Como a distribuição marginal pode ser representada na forma  $p(x_n) = \sum_{z_n} p(x_n, z_n)$ , segue-se que para cada ponto de dados observado  $x_n$

existe uma variável latente correspondente  $z_n$ . Usando o teorema de Bayes, pode-se obter a probabilidade condicional de  $z_{nc} = 1$  dado  $x_n$  como:

$$p(z_{nc} = 1|x_n) = \frac{p(z_{nc} = 1)p(x_n|z_{nc} = 1)}{\sum_{j=1}^C p(z_{nj} = 1)p(x_n|z_{nj} = 1)} = \frac{\pi_c p(x_n|\theta_c)}{\sum_{j=1}^C \pi_j p(x_n|\theta_j)} \quad (20)$$

onde  $\pi_c$  (isto é,  $p(z_{nc} = 1)$ ) é a probabilidade a priori de que o dado  $x_n$  foi gerado pelo componente  $c$  e  $p(z_{nc} = 1|x_n)$  é a probabilidade a posteriori de que o ponto  $x_n$  observado veio do componente  $c$ . A seguir, usa-se  $\gamma(z_{nc})$  para denotar  $p(z_{nc} = 1|x_n)$ , que também pode ser visto como a responsabilidade que o componente  $c$  assume para explicar a observação  $x_n$ .

Na formulação do modelo de mistura apresentada acima, precisa-se inferir um conjunto de parâmetros a partir da observação, incluindo as probabilidades de mistura  $\pi_c$  e os parâmetros para as distribuições de componentes  $\theta_c$ .

Pode-se escrever a probabilidade de gerar todos os pontos de dados da seguinte forma:

$$p(X|\Theta) = \prod_{n=1}^N \sum_{c=1}^C \pi_c p(x_n|\theta_c) \quad (21)$$

Na forma logaritmica, tem-se:

$$\ln p(X|\Theta) = \sum_{n=1}^N \ln \sum_{c=1}^C \pi_c p(x_n|\theta_c) \quad (22)$$

De acordo com [Bishop e Nasrabadi \(2006\)](#), a estimativa de máxima verossimilhança (MLE) é uma abordagem estatística importante para a estimativa de parâmetros e pode ser dada como:

$$\Theta_{MLE} = \arg \max_{\Theta} (\ln p(X|\Theta)) \quad (23)$$

O qual considera a melhor estimativa como aquela que maximiza a probabilidade de gerar todas as observações. Às vezes tem-se informações a priori  $p(\Theta)$  sobre os parâmetros que podem ser incorporadas aos modelos de mistura. Assim, a estimativa máxima a posteriori (MAP) é ([GAUVAIN; LEE, 1994](#); [SORENSEN, 1980](#)):

$$\Theta_{MAP} = \arg \max_{\Theta} (\ln p(X|\Theta) + \ln p(\Theta)) \quad (24)$$

Que considera a melhor estimativa como aquela que maximiza a probabilidade a posteriori de  $\Theta$  de acordo com os dados observados. MLE e MAP fornecem uma

abordagem unificada para estimativa de parâmetros, que é bem definida no caso da distribuição normal.

Como mencionado anteriormente, cada *cluster* é representado matematicamente por uma distribuição paramétrica. Em princípio, as misturas podem ser construídas com quaisquer tipos de componentes e ainda pode-se ter um modelo de mistura perfeitamente bom. Na prática, é comum considerar modelos paramétricos de mistura, onde todos os componentes são da mesma família paramétrica de distribuições, mas com parâmetros diferentes. Por exemplo, todos eles podem ser gaussianas com diferentes médias e variâncias.

### 3.5.1 Modelo de Misturas Gaussianas

O modelo de mistura mais conhecido é o modelo de mistura gaussiana (GMM), onde cada componente é uma distribuição gaussiana. Recentemente, o GMM tem sido amplamente utilizado para *Clustering* em muitas aplicações, como , segmentação de imagens e rastreamento de objetos (KARANAM; SRINIVAS; CHAKRAVARTY, 2023; ZHANG; JIN; GE, 2022).

De acordo com Yu *et al.* (2015), a distribuição gaussiana, também conhecida como distribuição normal, é um modelo amplamente utilizado para a distribuição de variáveis contínuas. No caso de uma única variável  $x$ , a distribuição gaussiana pode ser escrita na forma:

$$N(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}, \quad (25)$$

onde  $\mu$  é a média e  $\sigma^2$  é a variância. A Figura 8a mostra o gráfico de distribuições gaussianas para diferentes parâmetros. Para um vetor  $d$ -dimensional  $x$ , a distribuição gaussiana multivariada assume a forma:

$$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}} \frac{1}{|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (26)$$

onde  $\mu$  é um vetor médio  $d$ -dimensional,  $\Sigma$  é a  $d \times d$  matriz de covariância e  $|\Sigma|$  denota o determinante de  $\Sigma$ . A Figura 8b mostra contornos da distribuição gaussiana para vários valores de parâmetros.

No modelo de mistura gaussiana, cada componente é representado pelos parâmetros de uma distribuição gaussiana multivariada  $p(x_c|\theta_c) = N(x_n|\mu_c, \Sigma_c)$ . Com base na Equação 16, a distribuição da mistura gaussiana pode ser escrita como uma superposição linear de gaussianas na forma:

$$p(x_n|\Theta) = p(x_n|\pi, \mu, \Sigma) = \sum_{c=1}^C \pi_c N(x_n|\mu, \Sigma_c). \quad (27)$$

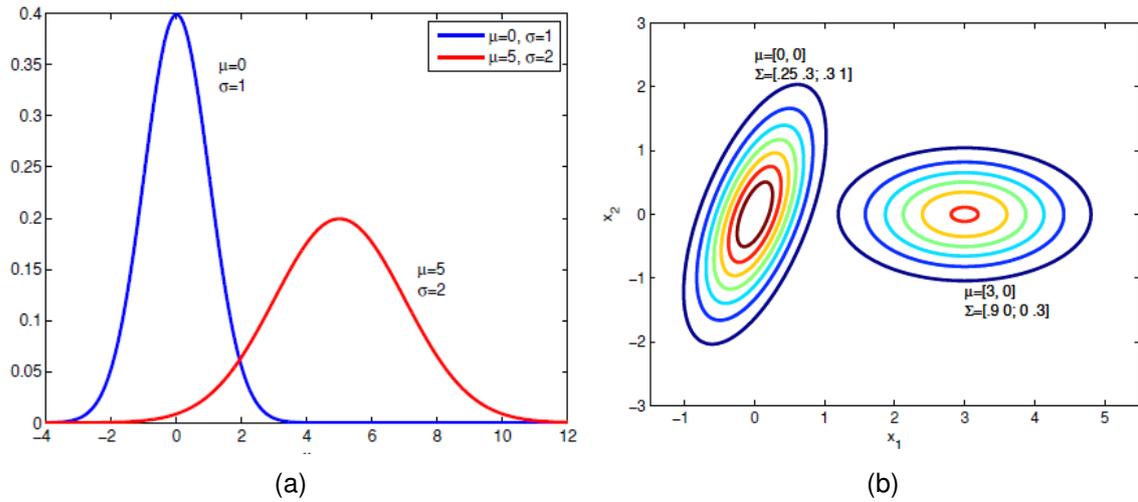


Figura 8 – (a) Gráficos da distribuição gaussiana univariada dada pela Equação 25 para vários parâmetros  $\mu$  e  $\sigma$ , e (b) contornos da distribuição gaussiana multivariada (2D) para vários parâmetros de  $\mu$  e  $\Sigma$ .

Fonte: Adaptado de D’Ortenzio e Manes (2021)

Para um dado conjunto de observações  $X$ , a função de log-verossimilhança é dada por:

$$l(\Theta) = \ln p(X|\Theta) = \sum_{n=1}^N \ln p(x_n|\Theta) = \sum_{n=1}^N \ln \sum_{c=1}^C \pi_c N(x_n|\mu_c, \Sigma_c) \quad (28)$$

Para encontrar soluções de máxima verossimilhança válidas em máximos locais, calcula-se as derivadas de  $\ln p(X|\pi, \mu, \Sigma)$  em relação a  $\pi_c$ ,  $\mu_c$  e  $\Sigma_c$ , respectivamente. A derivada em relação à média  $\mu_c$  é dada por:

$$\frac{\partial l}{\partial \mu_c} = \sum_{n=1}^N \frac{\pi_c N(x_n|\mu_c, \Sigma_c)}{\sum_{j=1}^C \pi_j N(x_n|\mu_j, \Sigma_j)} \Sigma_c^{-1} (x_n - \mu_c) = \sum_{n=1}^N \gamma(z_{nc}) \Sigma_c^{-1} (x_n - \mu_c), \quad (29)$$

Definindo esta derivada como zero e multiplicando por  $\Sigma_c$ , obtém-se as equações:

$$\mu_c = \frac{\sum_{n=1}^N \gamma(z_{nc}) x_n}{\sum_{n=1}^N \gamma(z_{nc})} \quad (30)$$

$$\gamma(z_{nc}) = \frac{\pi_c N(x_n|\mu_c, \Sigma_c)}{\sum_{j=1}^C \pi_j N(x_n|\mu_j, \Sigma_j)} \quad (31)$$

Pode-se interpretar que a média  $\mu_c$  para a  $c$ -ésima componente gaussiana é obtida tomando uma média ponderada de todos os pontos do conjunto de dados, em que o fator de ponderação corresponde à probabilidade a posteriori  $\gamma(z_{nc})$  de que a

componente  $c$  foi responsável por gerar  $x_n$ . Da mesma forma, define-se a derivada de  $\ln p(X|\pi, \mu, \Sigma)$  em relação a  $\Sigma_c$  como zero e obtém-se:

$$\Sigma_c = \frac{\sum_{n=1}^N \gamma(z_{nc})(x_n - \mu_c)(x_n - \mu_c)^T}{\sum_{n=1}^N \gamma(z_{nc})} \quad (32)$$

Semelhante a Equação 30, cada ponto de dados é ponderado pela probabilidade condicional gerada pelo componente correspondente e com o denominador dado pelo número efetivo de pontos associados ao componente correspondente.

A derivada de  $\ln p(X|\pi, \mu, \Sigma)$  em relação às probabilidades de mistura  $\pi_c$  requer um pouco mais de trabalho, porque os valores de  $\pi_c$  são obrigados a ser positivos e com soma igual a um. Essa restrição pode ser tratada usando um multiplicador de Lagrange  $\wp$  e maximizando a seguinte quantidade:

$$\ln p(X|\pi, \mu, \Sigma) + \wp \left( \sum_{c=1}^C \pi_c - 1 \right)$$

Depois de simplificar e reorganizar obtém-se:

$$\pi_c = \frac{\sum_{n=1}^N \gamma(z_{nc})}{N} \quad (33)$$

de modo que as probabilidades de mistura para o  $c$ -ésimo componente sejam dadas pela probabilidade a posteriori média que o componente assume para explicar os pontos de dados.

Vale ressaltar que as Equações 30, 32 e 33 não são a solução de forma fechada para os parâmetros do modelo de mistura. A razão é que essas equações estão intimamente ligadas à Equação 31. Mais especificamente, as probabilidades a posteriori  $\gamma(z_{nc})$  dadas pela Equação 31 dependem de todos os parâmetros do modelo de mistura, enquanto todos os resultados das Equações 30, 32 e 33 dependem de  $\gamma(z_{nc})$ . Portanto, maximizar a função log-verossimilhança para um modelo de mistura gaussiana torna-se um problema computacionalmente complexo. Um método elegante e poderoso para encontrar soluções de máxima verossimilhança para modelos com variáveis latentes é chamado de algoritmo de maximização de expectativa ou algoritmo EM (DEMPSTER; LAIRD; RUBIN, 1977). As Equações acima apresentam uma solução iterativa para o modelo de mistura gaussiana, que acaba por ser uma instância do algoritmo EM para o caso particular do modelo de mistura gaussiana.

O algoritmo EM começa inicializando com suposições sobre os parâmetros, incluindo as médias, covariâncias e probabilidades de mistura. Em seguida, alterna-se entre duas etapas de atualização, a etapa de expectativa e a etapa de maximização. Na etapa de expectativa, ou etapa  $E$ , usa-se os parâmetros atuais para calcular as probabilidades a posteriori de acordo com a Equação 31. Na etapa de maximização, ou etapa  $M$ , maximiza-se a log-verossimilhança com as probabilidades a posteriori

atualizadas e se re-estimam as médias, covariâncias e coeficientes de mistura usando as Equações 30, 32 e 33. Cada iteração do algoritmo EM tem a garantia de aumentar a log-verossimilhança. Na prática, o algoritmo EM converge quando a mudança na log-verossimilhança ou nos valores dos parâmetros cai abaixo de algum limite. O pseudocódigo do algoritmo EM para misturas gaussianas é apresentado no algoritmo 1.

---

### Algoritmo 1 EM para Misturas Gaussianas

---

**Entrada:** Um conjunto de pontos de dados e um modelo de mistura gaussiana

**Saída:** Maximizar a log-verossimilhança em relação aos parâmetros

**início**

Inicialize as médias  $\mu_c^0$ , covariâncias  $\Sigma_c^0$  e probabilidades de mistura  $\pi_c^0$  ;

**repita**

**E-step:** Calcule  $\gamma(z_{nc})$  usando os parâmetros atuais baseados na Equação 31 ;

**M-step: início**

Atualize as médias usando a Equação 30 ;

Utilize as novas médias para calcular as covariâncias usando a Equação 32 ;

Re-estime as probabilidades de mistura usando a Equação 33 ;

**fim**

Calcule a log-verossimilhança usando a Equação 28 ;

**se o critério de convergência for atendido então**

**retorne** os parâmetros finais  $\mu, \Sigma, \pi$  ;

**fim**

**até a convergência do algoritmo;**

**fim**

---

Conforme ilustrado na Figura 9, o algoritmo *EM* para uma mistura de dois componentes gaussianos é aplicado a um conjunto de dados gerados aleatoriamente. A Figura 9a mostra os pontos de dados em círculos azuis, juntamente com a inicialização aleatória do modelo de mistura no qual os dois componentes gaussianos são mostrados como círculos verdes e vermelhos. A Figura 9b mostra o resultado do passo *E* inicial, no qual cada ponto de dados é representado usando uma proporção de branco e preto de acordo com a probabilidade a posteriori de ter sido gerado pelo componente correspondente. A Figura 9c mostra a situação após o primeiro passo *M*, em que as médias e covariâncias de ambos os componentes mudaram. A Figura 9d mostra os resultados próximo a convergência após 47 ciclos de *EM*.

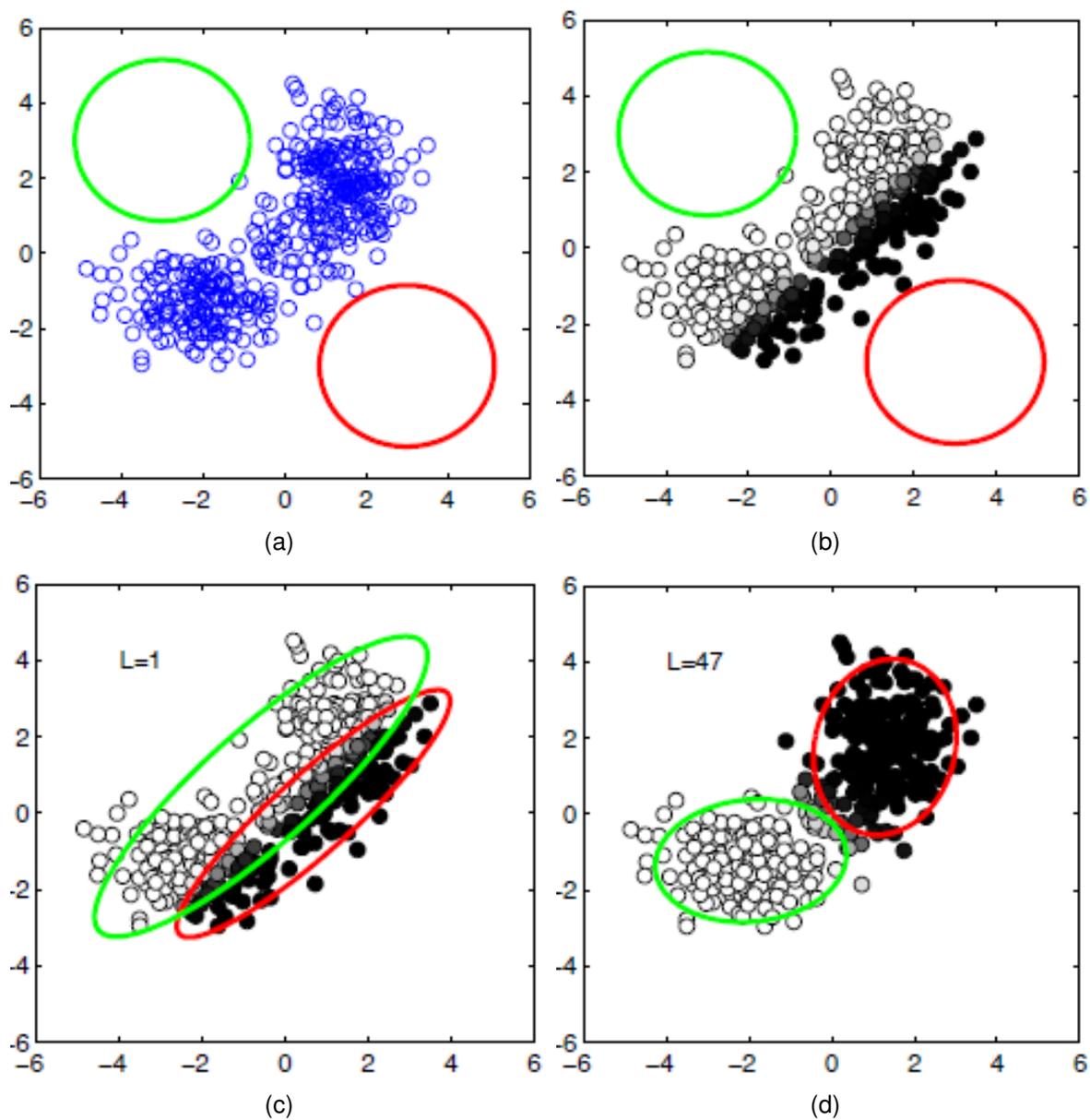


Figura 9 – Ilustração do algoritmo EM para duas componentes gaussianas.

Fonte: Adaptado de Bishop e Nasrabadi (2006)

## 4 MODELAGEM DE ESPAÇO DE BUSCA DE BAIXA DIMENSIONALIDADE

Neste capítulo, apresenta-se a modelagem de espaços de busca de baixa dimensionalidade, através da operacionalização das teorias discutidas no capítulo 3. A transição da teoria para a prática é efetuada através da exploração dos conceitos e fórmulas previamente introduzidos, proporcionando uma fundamentação sólida para o método LSMDE delineado no capítulo subsequente. A modelagem de baixa dimensionalidade não apenas simplifica a representação do espaço de busca, mas também facilita a identificação de padrões, a redução de complexidade computacional e a descoberta de *insights* valiosos para o domínio em estudo. Este processo é crucial para a criação de uma estrutura robusta que apoie a análise subsequente do método proposto.

Neste capítulo, apresentam-se as definições matemáticas e algorítmicas que regem o LSMDE. Algumas das formulações matemáticas já apresentadas no capítulo 3 de fundamentação teórica foram adaptadas para o contexto de Otimização de modo a facilitar o entendimento do funcionamento do método proposto.

### 4.1 PRINCIPIOS DE TRANSFORMAÇÃO DE ESPAÇO

Dada as formulações irrestritas ( $\chi = \mathbb{R}^d$ ) e restritas ( $\chi = [-1, 1]^d$ ) de Otimização Global em Larga Escala escritas tais como:

$$\begin{aligned} f_{IR}^* &= \min_{x \in \mathbb{R}^d} f(x) \\ f_R^* &= \min_{x \in \chi} f(x) \end{aligned} \tag{34}$$

onde  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  é uma função contínua e determinística.

Para um problema irrestrito (IR), assume-se que existe  $x^* \in \mathbb{R}^d$  tal que

$$\min_{x \in \mathbb{R}^d} f(x) = f(x^*) = f_{IR}^* \tag{35}$$

Isso sugere que a função  $f$  é limitada abaixo, ou seja,  $f_{IR}^* > -\infty$  e que o mínimo em (IR) é atingido (nem todos os minimizadores estão no infinito). Para restrito (R), esta propriedade está implícita no teorema do valor extremo de Weierstrass (PROTTER; CHARLES JR *et al.*, 2012), que diz que se  $f$  é uma função contínua definida sobre um compacto  $\chi$ ,  $f$  deve atingir o mínimo global em  $\chi$ .

Em uma tentativa de aliviar a maldição da dimensionalidade de problemas LSGO, foca-se nas aproximações de baixo posto de  $\chi$  (como descrito na secção 3.4). Alguns trabalhos relacionam essas funções com certos tipos de espaço denominados "subespaços efetivos", "subespaços ativos" ou "multi-ridge" (CONSTANTINE, 2015; VYBIRAL,

s.d.; TYAGI; CEVHER, 2014). Para facilitar a comunicação, esta tese utiliza o termo "espaço reduzido", sendo este derivado do cálculo da aproximação de baixo posto de  $\chi$ .

O subespaço reduzido permite diminuir a dimensão do espaço da função e com isso reduzir os custos computacionais de resolução do problema de otimização. Ilustre-se a ideia com o exemplo a seguir:

**Exemplo 4.1.1.** Considere o seguinte problema de otimização

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \text{sen}^2(x_1 - x_2 - 0,5) \\ \text{sujeito a} \quad & x \in \chi = [-2, 2]^2 \end{aligned} \quad (36)$$

Os mínimos globais do problema podem ser encontrados analiticamente. Resolvendo  $\text{sen}^2(x_1 - x_2 - 0,5) = 0$  descobre-se que o conjunto de minimizadores globais compreende os três conjuntos a seguir:

$$\begin{aligned} G_1^* &= \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} t - \begin{bmatrix} 0 \\ 0,5 - \pi \end{bmatrix} : -2 \leq t \leq 2,5 - \pi \right\} \\ G_2^* &= \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} t - \begin{bmatrix} 0 \\ 0,5 \end{bmatrix} : -1,5 \leq t \leq 2 \right\} \\ G_3^* &= \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} t - \begin{bmatrix} 0 \\ 0,5 + \pi \end{bmatrix} : -1,5 + \pi \leq t \leq 2 \right\}. \end{aligned} \quad (37)$$

Cada  $G_i^*$  corresponde a uma linha distinta de minimizadores globais ao longo da qual a função é constante e na qual atinge o seu mínimo global.

Pode-se reduzir a dimensão  $D$  do problema para uma dimensão  $D = 1$  parametrizando  $x_1 = y$  e  $x_2 = -y$ ; isso gera o seguinte problema unidimensional:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \text{sen}^2(2y - 0,5) \\ \text{sujeito a} \quad & [1, -1]^T y \in \chi \end{aligned} \quad (38)$$

As soluções são alcançadas nos pontos  $y^* = \pi_i/2 + 0,25$ ,  $i = 0, \pm 1$ . Geometricamente, a parametrização equivale a incorporação (*embedding*) da reta  $[1, -1]^T$  no espaço bidimensional e à busca dos mínimos globais ao longo desta reta. As soluções no espaço reduzido correspondem aos pontos de interseção entre  $[1, -1]^T$  e as linhas de minimizadores globais. As três soluções no espaço original são pontos  $x_i^* = (\pi k + 0,25) \cdot [1, -1]^T$  para  $i = 1, 2, 3$ .

## 4.2 PROBLEMAS REDUZIDOS VIA DECOMPOSIÇÃO DE VALOR SINGULAR

O Exemplo 4.1.1 ilustra que é possível considerar um problema de otimização como um problema de dimensão inferior, que tem o mesmo mínimo global  $f_{IR}^*(f^*)$ , restringindo-o e resolvendo (IR) e (R) apenas dentro deste importante subespaço. Normalmente, no entanto, o subespaço reduzido é desconhecido. Decomposição de Valor Singular é uma proposta para reduzir o tamanho do espaço original e, portanto, o custo de sua solução, enquanto tenta preservar os valores mínimos globais originais do problema. De maneira geral, nesta tese, investiga-se o problema reduzido restrito (RR):

$$\begin{aligned} & \underset{\psi}{\text{minimize}} && f(\psi) \\ & \text{sujeito a} && \psi \in \Psi \end{aligned} \tag{39}$$

onde  $\Psi = U_k S_k$  é um espaço reduzido resultante de uma aproximação de baixo posto  $k$  com  $k < d$  como descrito na subsecção 3.4.2; e  $\psi \in \Psi$  é uma solução no subespaço reduzido.

*Observação 1.* Se  $\psi^*$  é um minimizador para o problema reduzido restrito (RR), pode-se recuperar um correspondente minimizador  $x^*$  na dimensão original definindo  $x^* = \psi^* \cdot V_k^T$  (TOUGAS; SPITERI, 2007; WANG; JIN, 2006).

A otimização do problema reduzido restrito (RR) é bem-sucedida se seus mínimos globais coincidem com o mínimo global do problema reduzido (R) dentro da tolerância  $\epsilon$  definido pelo lema de Johnson-Lindenstrauss e descrito na subsecção 3.4.3.

**Definição 4.2.1** (Problema reduzido restrito bem-sucedido). Dizemos que o problema reduzido restrito (RR) é bem-sucedido se existe um  $\psi^* \in \mathbb{R}^k$  tal que o seu correspondente  $x^* \in \mathbb{R}^d$  também é o minimizador do problema reduzido (R) respeitando a tolerância  $\epsilon$ .

## 4.3 TRANSFORMAÇÃO DE ESPAÇO

**Definição 4.3.1** (Transformação de espaço bem-sucedida). Uma função  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  tem um espaço reduzido de dimensão  $k$  quando existe um subespaço  $\Psi$  de dimensão  $k < d$  tal que para todos os vetores  $\psi \in \Psi$ , tem-se que a função de aptidão  $f(\psi) \simeq f(x)$  dado a tolerância  $\epsilon$  e  $x \in \chi$ . O subespaço  $\Psi$  é denominado espaço reduzido de  $\chi$ .

Pela aplicação de SVD em (IR) e (R), deixa-se de otimizar  $\mathbb{R}^d$  para otimizar em um espaço de dimensão inferior  $\mathbb{R}^k$ . A relação  $\psi \rightarrow x V_k S_k^{-1}$  mapeia pontos de  $\mathbb{R}^d$  para pontos ao longo do subespaço reduzido em  $\mathbb{R}^k$ .

**Teorema 4.1.** *Pela definição de SVD, dado que  $A = USV^T$ . Então para qualquer*

$x \in \mathbb{R}^d$ , existe um  $\psi \in \mathbb{R}^k$  e  $\psi = xV_kS_k^{-1}$  tal que  $f(x) \simeq f(\psi)$ . Em particular, para um minimizador global  $x^*$  de  $(IR)$  ou  $(R)$ , existe um  $\psi^* \in \mathbb{R}^k$  tal que  $f(\psi^*) \simeq f(x^*)$ .

O teorema 4.1 estabelece que existe um minimizador  $\psi^* \in \mathbb{R}^k$  pelo qual o correspondente valor  $x^* \in \mathbb{R}^d$  é um minimizador. Assim, é possível recuperar um minimizador global através da busca em  $\mathbb{R}^k$ .

#### 4.4 MODELAGEM DE ESPAÇO DE BUSCA REDUZIDO $\Psi$

Nem todas as incorporações (*embeddings*) podem recuperar com exatidão o mínimo global do problema original. Portanto, espera-se que os problemas reduzidos sejam bem sucedidos para a matriz  $\Psi$  dentro de uma margem de tolerância que está correlacionada com o parâmetro  $\epsilon$ , a dimensionalidade  $k$  e as características intrínsecas do espaço de busca original. A partir disso, é necessário tomar medidas para garantir que o ruído ocasionado pela transformação de espaço  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^k$  não interfira na exploração do subespaço reduzido  $\Psi$ .

Uma forma de guiar a exploração no espaço reduzido  $\Psi$  é analisar cada  $\psi \in \Psi$  na perspectiva de *clusters*, de forma que progresso para a solução ótima não seja definido apenas por uma solução única  $\psi_i$ , mas sim pelo *cluster* que representa todas as soluções  $\psi$  que possuem características semelhantes a  $\psi_i$ . A Figura 10 ilustra como seria um processo de *Clustering*.

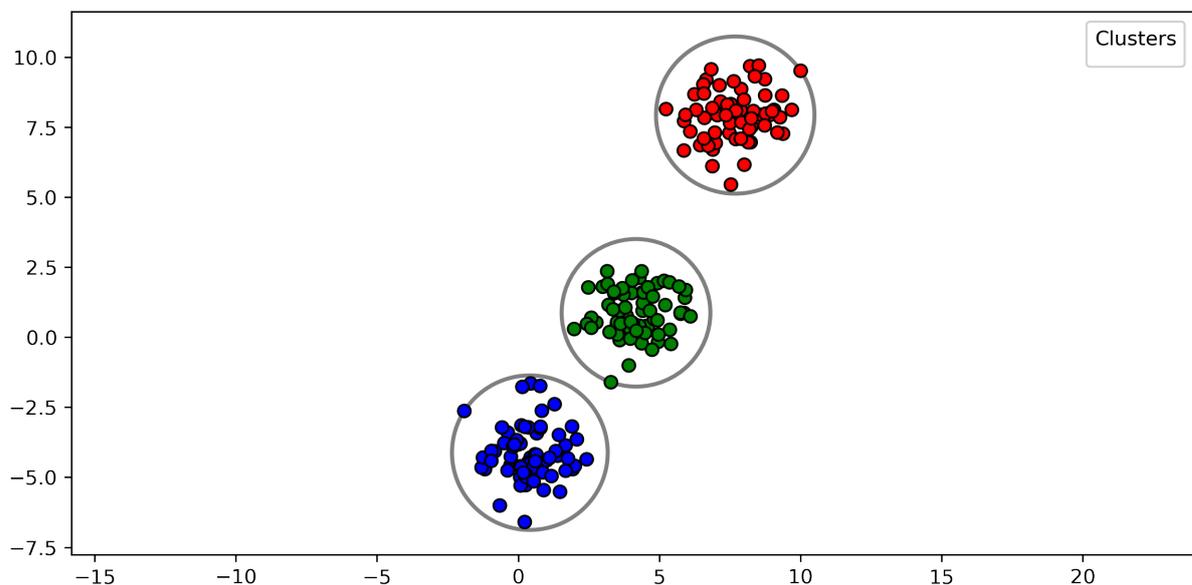


Figura 10 – Ilustração do processo de *Clustering* em um espaço reduzido  $\Psi$  2D utilizando *k*-means

Fonte: Autor

Contudo, devido a perda de informação ocasionada pela transformação de espaço  $\phi$ , podem existir soluções  $\psi$  que possuem similaridades a mais de um *cluster*. Logo, essa incerteza deve ser mensurada e utilizada para guiar a exploração no espa-

ço reduzido. A Figura 11 ilustra um processo de *Clustering* por modelos de misturas gaussianas no espaço reduzido onde as soluções possuem similaridades a mais de um *cluster*.

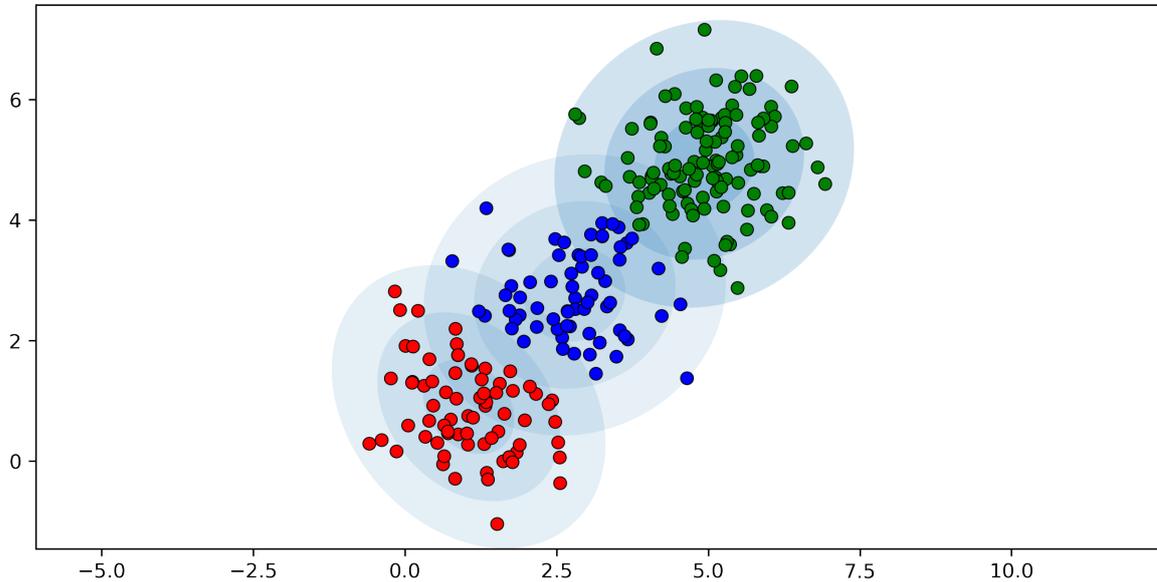


Figura 11 – Ilustração de clusterização em um espaço reduzido  $\Psi$  2D utilizando modelo de misturas gaussianas

Fonte: Autor

Segundo a definição de misturas gaussianas explicitado na subsecção 3.5.1, cada *cluster* é representado por um conjunto de parâmetros  $\theta = [\pi, \mu, \Sigma]$ , em que

- A média  $\mu$  define o centro do *cluster*.
- A covariância  $\Sigma$  define sua largura. Isto seria equivalente às dimensões de um elipsoide num cenário de alta dimensionalidade (LSGO).
- Uma probabilidade de mistura  $\pi$  que define quão representativa é a gaussiana em relação ao espaço de busca. Quanto maior o  $\pi$  do *cluster*, mais soluções  $\psi_i$  podem ser explicadas pelo mesmo.

Na Figura 12, cada gaussiana explica os dados contidos em cada um dos três *clusters* disponíveis. Os coeficientes de mistura são probabilidades e devem atender a condição  $\sum_1^k \pi_k = 1$ .

Para determinar os valores ideais para  $\theta$  deve-se garantir que cada gaussiana se ajuste aos pontos de dados pertencentes a cada *cluster*. Isso pode ser alcançado através da máxima verossimilhança. Na perspectiva de um espaço de busca de um problema de otimização, a função densidade gaussiana é dada pela Equação 40:

$$\mathcal{N}(\psi|\mu, \Sigma) = \frac{1}{(2\pi)^{k/2}} \frac{1}{|\Sigma|^{1/2}} e^{-\frac{1}{2}(\psi-\mu)^T \Sigma^{-1}(\psi-\mu)} \quad (40)$$

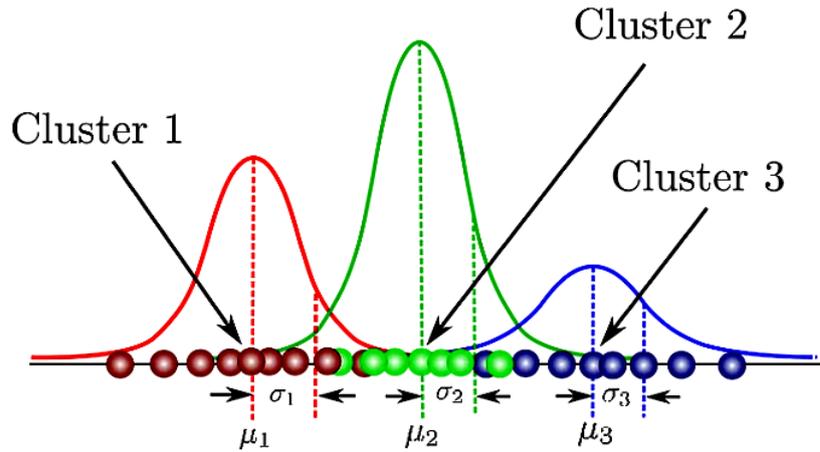


Figura 12 – Ilustração de clusterização em um espaço reduzido  $\Psi$  2D utilizando modelo de misturas gaussianas

Fonte: Autor

Onde  $\psi$  representa uma solução no espaço de busca reduzido,  $k$  é o número de dimensões do espaço reduzido e  $\mu$  e  $\Sigma$  são a média e a covariância, respectivamente. Sob um ponto de vista de um problema LSGO, dado um conjunto de soluções com 1000 dimensões ( $k = 1000$ ) composto por  $N = 500$  soluções candidatas, então o subespaço  $\Psi$  será uma matriz  $500 \times 1000$  onde  $\mu$  será um vetor  $1 \times 1000$  e  $\Sigma$  será uma matriz  $1000 \times 1000$ . É importante analisar o logaritmo da Equação 26, que é dado por:

$$\ln N(\psi|\mu, \Sigma) = -\frac{k}{2} \ln 2\pi - \frac{1}{2} \ln \Sigma - \frac{1}{2}(\psi - \mu)^T \Sigma^{-1}(\psi - \mu) \quad (41)$$

Diferenciar a Equação 41 em relação à média e à covariância e depois igualá-la a zero permite encontrar os valores ótimos para estes parâmetros, e as soluções corresponderão às Estimativas de Máxima Verossimilhança (MLE) para esta configuração.

Para que o processo de otimização ocorra de maneira efetiva no espaço reduzido, deve-se expressar qual a probabilidade de uma determinada solução candidata  $\psi_i$  vir de um *cluster* (ou gaussiana)  $\lambda$ , Pode-se expressar isso como:

$$p(z_{i,\lambda} = 1|\psi_i) \quad (42)$$

Neste caso,  $z$  é uma variável latente que assume apenas dois valores possíveis. É igual a 1 quando  $\psi$  pode ser explicado pelo cluster  $\lambda$  e zero caso contrário. Da mesma forma, pode-se afirmar que:

$$\pi_\lambda = p(z_\lambda = 1) \quad (43)$$

O que significa que a probabilidade geral de observar uma solução que vem do cluster  $\lambda$  é equivalente ao coeficiente de mistura dessa gaussiana. Ou seja, quanto

maior a amplitude da gaussiana, maior será essa expectativa de probabilidade. Seja  $Z$  o conjunto de todas as variáveis latentes possíveis  $z$ , tem-se:

$$Z = z_1, \dots, z_\lambda \quad (44)$$

Sabe-se que cada  $z$  ocorre independentemente dos outros e que só podem assumir o valor igual a 1 quando  $\lambda$  for igual ao *cluster* de onde provém a solução. Portanto:

$$p(z) = p(z_1 = 1)^{z_1} p(z_2 = 1)^{z_2} \dots p(z_\lambda = 1)^{z_\lambda} = \prod_{\lambda=1}^{\Lambda} \pi_\lambda^{z_\lambda} \quad (45)$$

Para encontrar a probabilidade de observar as soluções  $\psi_i$ , visto que eles vieram do cluster  $\lambda$  usa-se a própria função gaussiana como segue:

$$p(\psi_i|z) = \prod_{\lambda=1}^{\Lambda} \mathcal{N}(\psi_i|\mu_\lambda, \Sigma_\lambda)^{z_\lambda} \quad (46)$$

Objetiva-se inicialmente determinar qual a probabilidade de  $z$  dada a observação  $\psi_i$ . As equações derivadas acima juntamente com a regra de Bayes, auxiliam a determinar esta probabilidade. Pela regra do produto de probabilidades, sabe-se que:

$$p(\psi_i, z) = p(\psi_i|z)p(z) \quad (47)$$

Como pode-se observar, as derivações das Equações 45 e 46 são os operandos à direita da Equação 47. Para usar a regra de Bayes para obter a probabilidade primeiro precisa-se de  $p(\psi_i)$ , não de  $p(\psi_i, z)$ . Para remover  $z$  pode-se usar Marginalização, somando os termos de  $z$ , portanto:

$$p(\psi_i) = \sum_{\lambda=1}^{\Lambda} p(\psi_i|z)p(z) = \sum_{\lambda=1}^{\Lambda} \pi_\lambda \mathcal{N}(\psi_i|\mu_\lambda, \Sigma_\lambda) \quad (48)$$

A Equação 48 define uma mistura gaussiana, e como tal, depende de  $\theta = [\pi, \mu, \Sigma]$ . Para determinar os valores ideais para  $\theta$ , precisa-se determinar a máxima verossimilhança do modelo.

O espaço de otimização reduzido pode ser modelado como a probabilidade conjunta de todas as observações  $\psi_i$ , definida por:

$$p(\Psi) = \prod_{i=1}^N p(\psi_i) = \prod_{i=1}^N \sum_{\lambda=1}^{\Lambda} \pi_\lambda \mathcal{N}(\psi_i|\mu_\lambda, \Sigma_\lambda) \quad (49)$$

Como feito na função de densidade gaussiana original, aplica-se o logaritmo a cada lado da equação de forma a ter-se:

$$\ln p(\Psi) = \sum_{i=1}^N \ln \sum_{\lambda=1}^{\Lambda} \pi_{\lambda} \mathcal{N}(\psi_i | \mu_{\lambda}, \Sigma_{\lambda}) \quad (50)$$

Para que se possa encontrar as gaussianas (*clusters*) que melhor explicam o espaço de busca reduzido, deve-se encontrar os parâmetros  $\theta$  que as descrevem. Para estimar os parâmetros ótimos  $\theta$  para a mistura, pode-se usar um método iterativo. Mas primeiro, deve-se encontrar a probabilidade de  $z$  dado  $\psi$ . A partir da regra de Bayes, tem-se:

$$p(z_{\lambda} = 1 | \psi_i) = \frac{p(\psi_i | z_{\lambda} = 1)p(z_{\lambda} = 1)}{\sum_{j=1}^{\Lambda} p(\psi_i | z_j = 1)p(z_j = 1)} \quad (51)$$

Substituindo as Equações 43, 48 na Equação 51, tem-se que:

$$p(z_{\lambda} = 1 | \psi_i) = \frac{\pi_{\lambda} \mathcal{N}(\psi_i | \mu_{\lambda}, \Sigma_{\lambda})}{\sum_{j=1}^{\Lambda} \pi_j \mathcal{N}(\psi_i | \mu_j, \Sigma_j)} = \gamma(z_{i\lambda}) \quad (52)$$

Onde  $\gamma(z_{i\lambda})$  representa a probabilidade a posteriori que cada *cluster*  $\lambda$  tem pela observação de  $\psi$ .

O algoritmo de maximização de expectativa, ou simplesmente algoritmo EM, é amplamente utilizado para problemas de otimização onde a função objetivo tem complexidades como a encontrada para o caso de modelo de misturas gaussianas.

Sejam os parâmetros do modelo de mistura gaussiana que explicam o espaço reduzido  $\Psi$  dado por:

$$\theta = [\pi, \mu, \Sigma] \quad (53)$$

Para se usar o algoritmo EM para encontrar os melhores parâmetros primeiro inicializa-se  $\theta$  aleatoriamente. O passo seguinte, denominado "Expectativa", avalia:

$$Q(\theta^*, \theta) = \mathbb{E}[\ln p(\Psi, Z | \theta^*)] = \sum_Z p(Z | \Psi, \theta) \ln p(\Psi, Z | \theta^*) \quad (54)$$

O termo  $p(Z | \Psi, \theta)$  é o mesmo  $\gamma(z_{i\lambda})$  apresentado na Equação 52. Para modelos de mistura gaussiana, a etapa de expectativa se resume ao cálculo do valor de  $\gamma$  usando os valores antigos dos parâmetros. Substituindo a Equação 52 em 54, tem-se:

$$Q(\theta^*, \theta) = \sum_Z \gamma(z_{i\lambda}) \ln p(\Psi, Z | \theta^*) \quad (55)$$

O termo  $p(\Psi, Z|\theta^*)$  pode ser encontrado usando a seguinte expressão:

$$p(\Psi, Z|\theta^*) = \prod_{i=1}^N \prod_{\lambda=1}^{\Lambda} \pi^{z_{i\lambda}} \mathcal{N}(\psi_i|\mu_\lambda, \Sigma_\lambda)^{z_{i\lambda}} \quad (56)$$

Em que a Equação 56 é o resultado do cálculo da probabilidade conjunta de todas as observações e variáveis latentes e é uma extensão das derivações iniciais para  $p(\psi)$ . O logaritmo dessa expressão é dado por:

$$\ln p(\Psi, Z|\theta^*) = \sum_{i=1}^N \sum_{\lambda=1}^{\Lambda} z_{i\lambda} [\ln \pi_\lambda + \ln \mathcal{N}(\psi_i|\mu_\lambda, \Sigma_\lambda)] \quad (57)$$

Para remover a variável latente  $Z$ , pode-se substituir a Equação 57 em 55, obtendo-se:

$$Q(\theta^*, \theta) = \sum_{i=1}^N \sum_{\lambda=1}^{\Lambda} \gamma(z_{i\lambda}) [\ln \pi_\lambda + \ln \mathcal{N}(\psi_i|\mu_\lambda, \Sigma_\lambda)] \quad (58)$$

Para estimar os parâmetros  $\theta$  deve-se maximizar  $Q$  em relação aos parâmetros  $\pi, \mu, \Sigma$ . Na etapa de "Maximização", encontra-se os parâmetros da mistura. Para tanto, precisa-se fazer de  $Q$  um problema de maximização restrita e assim adicionar um multiplicador de Lagrange  $\wp$  na Equação 58.

De modo geral, deseja-se atualizar  $\theta^*$  usando:

$$\theta^* = \arg \max_{\theta} Q(\theta^*, \theta)$$

Em que  $Q(\theta^*, \theta)$  é dado pela Equação 58. No entanto,  $Q$  também deve levar em conta a restrição de que todos os valores de  $\pi$  devem somar 1. Para fazer isso, precisa-se adicionar um multiplicador de Lagrange  $\wp$  adequado. Portanto, deve-se reescrever a Equação 58 na seguinte forma:

$$Q(\theta^*, \theta) = \sum_{i=1}^N \sum_{\lambda=1}^{\Lambda} \gamma(z_{i\lambda}) [\ln \pi_\lambda + \ln \mathcal{N}(\psi_i|\mu_\lambda, \Sigma_\lambda)] - \wp \left( \sum_{\lambda=1}^{\Lambda} \pi_\lambda - 1 \right) \quad (59)$$

Deste modo, pode-se determinar os parâmetros  $\theta$  a partir da máxima verossimilhança usando a derivada de  $Q$  em relação a  $\pi, \mu, \Sigma$  e igualá-las a zero separadamente obtendo-se as atualizações:

$$\pi_\lambda^* = \frac{\sum_{i=1}^N \gamma(z_{i\lambda})}{N}, \quad \mu_\lambda^* = \frac{\sum_{i=1}^N \gamma(z_{i\lambda}) \psi_i}{\sum_{i=1}^N \gamma(z_{i\lambda})}, \quad \Sigma_\lambda^* = \frac{\sum_{i=1}^N \gamma(z_{i\lambda}) (\psi_i - \mu_\lambda) (\psi_i - \mu_\lambda)^T}{\sum_{i=1}^N \gamma(z_{i\lambda})} \quad (60)$$

Em que:

$$\gamma(z_{i\lambda}) = \frac{\pi_\lambda \mathcal{N}(\psi_i | \mu_\lambda, \Sigma_\lambda)}{\sum_{j=1}^{\Lambda} \pi_j \mathcal{N}(\psi_i | \mu_j, \Sigma_j)} \quad (61)$$

Os valores revisados  $\theta^* = [\pi^*, \mu^*, \Sigma^*]$  são utilizados para determinar  $\gamma$  na próxima iteração EM e assim por diante até obter-se a convergência no valor de verossimilhança. A Equação 50 é utilizada para monitorar a probabilidade logarítmica em cada etapa de modo a atingir um máximo local.

## 5 PROPOSTA DA TESE

### 5.1 DESCRIÇÃO DO MÉTODO PROPOSTO (LSMDE)

*Low-dimensional Space Modeling-based Differential Evolution* ou LSMDE procura dividir o espaço de busca reduzido via Decomposição de Valor Singular, localizando regiões promissoras por meio do agrupamento dessas soluções em *clusters*. O LSMDE define um *cluster*  $\lambda$  a partir do parâmetro  $\theta = (\mu, \pi, \Sigma)$ . O centro  $\mu_i$  é uma solução no espaço reduzido que representa o *cluster*  $\lambda_i$ , identificando a sua localização dentro do espaço de busca reduzido. O coeficiente de mistura  $\pi_i$ , que especifica a probabilidade de que uma dada solução ter sido gerada pelo *cluster*, e covariância  $\Sigma$  que especifica as dimensões do *cluster*. Ao invés de armazenar todas as soluções agrupadas no *cluster*, as informações do grupo de soluções são explicadas através da gaussiana que representa o *cluster*. Um *cluster* se torna promissor quando o coeficiente de mistura atingir certo limitante  $\rho$ . Caso um *cluster* seja definido como promissor, uma busca local especializada em LSGO é aplicada para explorar apenas a região de alta dimensionalidade referente ao *cluster* escolhido, reduzindo o custo computacional associado com a tarefa.

O LSMDE é um método iterativo que possui quatro componentes principais: inicialização da população, transformação de espaço, otimização por modelos de mistura gaussiana e busca local. A Figura 13 apresenta uma visão de alto nível do método, destacando suas etapas principais em cinza.

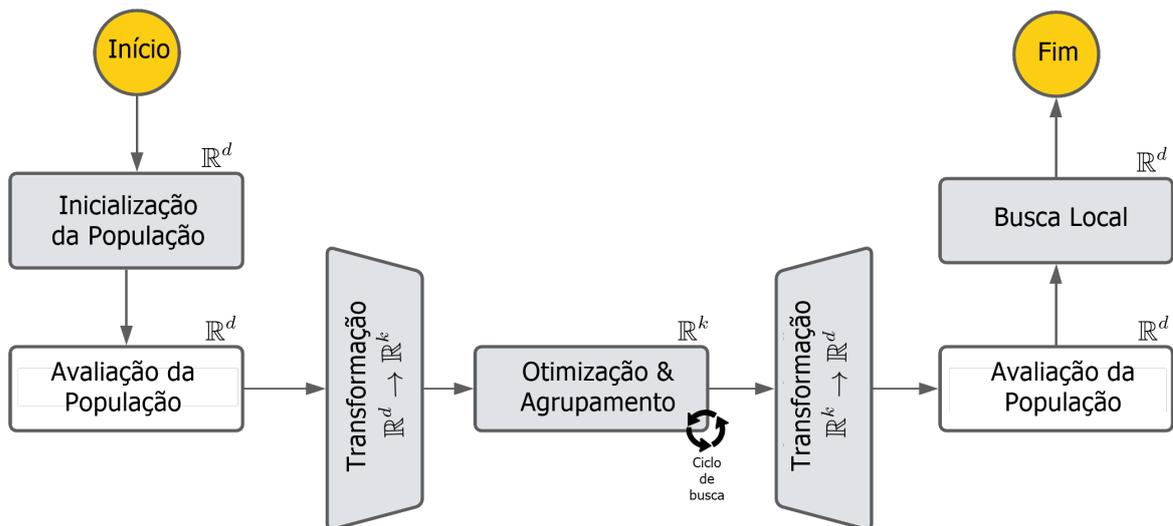


Figura 13 – Visão de alto nível do método LSMDE. As etapas principais estão destacadas em cinza

Fonte: Autor

A cada iteração do LSMDE, uma população  $P$  de soluções candidatas ainda em alta dimensionalidade é gerada pela metaheurística e enviada para o processo de redução de dimensionalidade de modo a construir o espaço reduzido atual  $\Psi$ . Posterior-

mente, as soluções  $\psi \in \Psi$  são enviadas para o processo de *Clustering*. Essas soluções são então agrupadas nos *clusters*  $\Lambda$  via modelo de misturas gaussianas. Novas soluções, decorrentes do processo natural de exploração da metaheurística, atualizam os centros dos *clusters* ( $\mu$ ) por meio de um processo de assimilação, fazendo com que os centros  $\mu$  se desloquem no espaço de busca reduzido. Em seguida é analisado o coeficiente de mistura  $\pi$  dos *clusters*. Caso algum dos coeficientes de mistura tenha atingido um limitante  $\rho$ , esse *cluster* pode estar em uma região de busca promissora. As subseções a seguir detalham esse processo.

### 5.1.1 Inicialização da População

A primeira etapa para a solução de um problema LSGO é definir a população que dará início ao processo de otimização. Em geral, uma boa população inicial melhora o desempenho dos algoritmos e pode economizar recursos computacionais durante o processo de busca (MAHDAVI; RAHNAMAYAN; DEB, 2018). A inicialização aleatória ou uniforme da população está entre as técnicas mais comuns. No entanto, essas técnicas não são recomendadas para problemas em grande escala (KAZIMIPOUR; LI; QIN, 2014). Aprendizado baseado em oposição (OBL) tem despertado interesse na última década, principalmente por causa de seu amplo uso em algoritmos de *soft computing* (MAHDAVI; RAHNAMAYAN; DEB, 2018). Como definido na Seção 3.3, o conceito computacional de aprendizado por oposição foi inspirado pelo conceito de número oposto ou candidato oposto, definido a seguir:

**Definição 5.1.1** (Candidato Oposto no espaço  $d$ -dimensional). Seja  $x = (x_1, \dots, x_d)$  um ponto no espaço  $d$ -dimensional e  $x_j \in [a_j, b_j], j = 1, 2, \dots, d$ . O oposto de  $\hat{x}$  é definido por  $\hat{x} = (\hat{x}_1, \dots, \hat{x}_d)$  como segue:

$$\hat{x}_j = a_j + b_j - x_j \quad (62)$$

De acordo com Mahdavi, Rahnamayan e Deb (2018), a busca por soluções ótimas considerando a aleatoriedade e seu oposto fornece uma probabilidade maior de encontrar regiões promissoras, uma vez que a busca pode ser redirecionada para regiões mais favoráveis em direções opostas. LSMDE usa um aprendizado baseado em oposição parcial modificado com três diferentes operadores de oposição para cada dimensão do problema, conforme definido a seguir.

**Definição 5.1.2** (Estratégia de Oposição do LSMDE). Seja  $x = (x_1, \dots, x_d)$  uma solução no espaço  $d$ -dimensional e  $x_j \in [a_j, b_j], j = 1, 2, \dots, d$ . O opositor  $\hat{x}$  é definido por  $\hat{x} = (\hat{x}_1, \dots, \hat{x}_d)$  onde cada  $\hat{x}_j$  é selecionado aleatoriamente entre os três operadores de

oposição seguintes:

$$\hat{x}_j = \begin{cases} a_j + b_j - x_j \\ x_j + a_j - ((a_j + b_j)/2) \bmod (b_j - a_j) \\ rand(a_j + b_j - x_j, (a_j + b_j)/2) \end{cases} \quad (63)$$

Geralmente, os esquemas de oposição tradicionais calculam a oposição para todas as variáveis de uma solução candidata usando o mesmo operador de oposição (MAHDAVI; RAHNAMAYAN; DEB, 2018). Essa técnica limita a diversidade de soluções opostas, já que o "espaço oposto" sempre segue a mesma regra de formação. A adição aleatória de operadores de oposição permite a construção de soluções opostas com uma variabilidade mais significativa em cada dimensão, permitindo explorar regiões do espaço de busca de alta dimensionalidade que seriam difíceis de alcançar apenas por meio de um algoritmo evolutivo. Portanto, o objetivo é gerar uma população inicial com alta diversidade que permita uma exploração mais eficiente do espaço de busca  $d$ -dimensional nas etapas seguintes de otimização.

Dado um problema de minimização (RR) em alta dimensionalidade, uma população de indivíduos  $x_i \in P$  gerada aleatoriamente e sua população oposta  $\hat{x}_i \in \hat{P}$  gerada de acordo com a Equação 63, o LSMDE gera cada novo indivíduo  $x_i$  da nova população inicial  $P$  da seguinte forma:

$$x_i = \min(f(x_i), f(\hat{x}_i)) \mid \forall x_i, \hat{x}_i \in (P \cup \hat{P}) \text{ e } i = 1, 2, \dots, n \quad (64)$$

onde  $f$  é uma função de aptidão e  $n$  é o número de indivíduos na população. Em outras palavras, a população inicial é formada pelos melhores indivíduos entre a população aleatória  $P$  e a população oposta  $\hat{P}$ . Antes de iniciar a etapa de otimização, o LSMDE reduz a dimensionalidade da população inicial gerada em  $P$  a fim de reduzir a complexidade do problema geral de Otimização em Grande Escala.

### 5.1.2 Transformação de espaço

Dada qualquer população  $P$  como um subconjunto do espaço de busca original de alta dimensionalidade, o LSMDE representa esse subconjunto como uma matriz  $P^{n \times d}$ , onde  $n$  é o número de indivíduos na população e  $d$  é a dimensionalidade do problema. De acordo com a Decomposição em Valores Singulares (SVD), qualquer matriz  $P^{n \times d}$  pode ser decomposta em três matrizes  $U, S, V$ , ilustradas na Figura 14, onde  $V^T$  é a transposta da matriz  $V$ , conforme segue:

$$P = U \cdot S \cdot V^T \quad (65)$$

- $S$  é uma matriz diagonal  $d \times d$  com  $s_{11} \geq s_{22} \geq \dots s_{dd} \geq 0$ , chamada de valores singulares.
- $U$  é uma matriz  $n \times d$  chamada de vetores singulares à esquerda, onde  $U^T \cdot U = I$ .

- $V$  é uma matriz  $d \times d$  chamada de vetores singulares à direita, onde  $V \cdot V^T = V^T \cdot V = I$ .

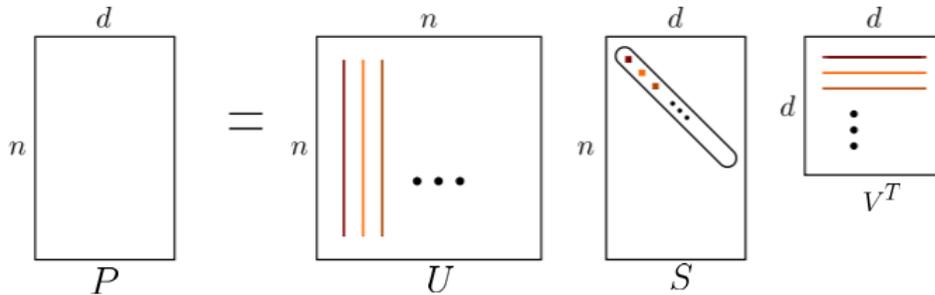


Figura 14 – Decomposição de matriz

Fonte: (FONSECA *et al.*, 2022)

Podemos reescrever a Equação 65 como segue:

$$P = \sum_{i=1}^{\min(n,d)} s_i \cdot \vec{u}_i \cdot \vec{v}_i^T \tag{66}$$

onde  $s_i$  é o  $i$ -ésimo valor singular e  $u_i, v_i^T$  são os correspondentes vetores singulares à esquerda e direita transpostos. Em outras palavras, a SVD expressa o subespaço de alta dimensionalidade  $P$  como uma combinação linear não negativa de  $\min(n, d)$  matrizes de posto 1, com os valores singulares fornecendo os multiplicadores e os produtos externos dos vetores singulares à esquerda e direita fornecendo as matrizes de posto 1. Como a SVD de  $P$  representa o subespaço  $P$  como uma multiplicação de matrizes ordenadas por importância, pode-se manter apenas os  $k$  vetores singulares mais importantes para obter uma projeção  $US \in \mathbb{R}^k$  do subespaço  $P \in \mathbb{R}^d$ , onde  $k < d$ . Em outras palavras, a SVD reduz a dimensionalidade da matriz  $P$  mantendo apenas os primeiros  $k$  termos na Equação 66 gerando assim uma aproximação de baixo posto  $\tilde{P}$  similar ao original  $P$ , conforme definido a seguir:

$$\tilde{P} = \sum_{i=1}^k s_i \cdot u_i \cdot v_i^T \text{ ou } \tilde{P} = U_k \cdot S_k \cdot V_k^T \tag{67}$$

A matriz  $\Psi = U_k S_k$  fornece representações de baixa dimensão do espaço de busca  $P$  a partir dos seus  $k$  componentes principais mais importantes (PC), e a matriz  $V_k^T$  pode ser interpretada como uma matriz de reconstrução que projeta essas soluções de baixa dimensão de volta ao espaço aproximado de alta dimensão  $\tilde{P}$  (Figura 15).

Para que o subespaço  $\Psi$  seja utilizado em uma abordagem baseada em população, é necessário desenvolver uma maneira de atualizar o subespaço de baixa dimensão sem ter que reconstruí-lo. Essa funcionalidade é importante para que os operadores de mutação, cruzamento e seleção da metaheurística possam ser utilizados de maneira efetiva. Para essa tarefa, pode-se utilizar as definições apresentadas na subsecção 4.3.

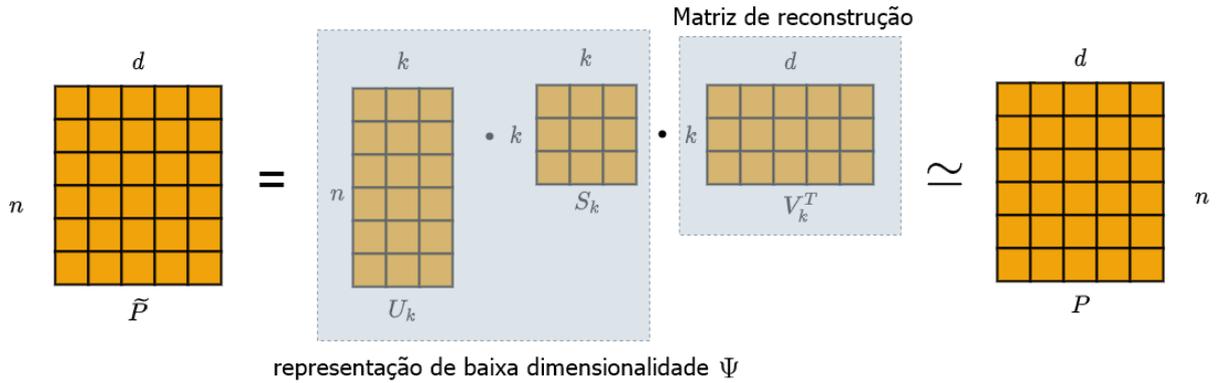


Figura 15 – Ilustração de uma representação de baixa dimensão de uma matriz aleatória  $P$  de  $d$  dimensões para uma matriz  $\Psi$  de  $k$  dimensões.

Fonte: (FONSECA *et al.*, 2022)

Para uma aproximação do subespaço  $\tilde{P}$ , as três matrizes decompostas  $U_k$ ,  $S_k$  e  $V_k$  são calculadas primeiro. No entanto, quando um novo indivíduo é adicionado ao espaço de busca por meio de uma abordagem baseada em população, não é necessário reiniciar o cálculo do subespaço de baixa dimensão  $\Psi$ . Pode-se usar os conceitos apresentados anteriormente para construir  $\Psi^{(t+i)}$  como um processo incremental (TOUGAS; SPITERI, 2007; WANG; JIN, 2006).

Seja  $U_k S_k$  a matriz que compõe a aproximação de baixo posto do subespaço  $P \in \mathbb{R}^{n \times d}$ , dado um novo indivíduo  $x_i \in \mathbb{R}^{n \times d}$  para ser projetado em um espaço  $k$ -dimensional  $\Psi$ , tem-se  $u = x_i \cdot V_k \cdot S_k^{-1}$ . A projeção  $u \in \mathbb{R}^{n \times k}$  é mesclada na decomposição existente através da sua adição a parte inferior da matriz  $U_k$ , resultando em uma matriz  $U_k^{(t+1)} \in \mathbb{R}^{(n+1) \times k}$ . Essa nova matriz pode ser usada para calcular  $\Psi^{(t+1)} = U_k^{(t+1)} \cdot S_k$  para a próxima geração da metaheurística baseada em população. A Figura 16 ilustra a projeção de um novo indivíduo  $x_i$  em  $\Psi$ .

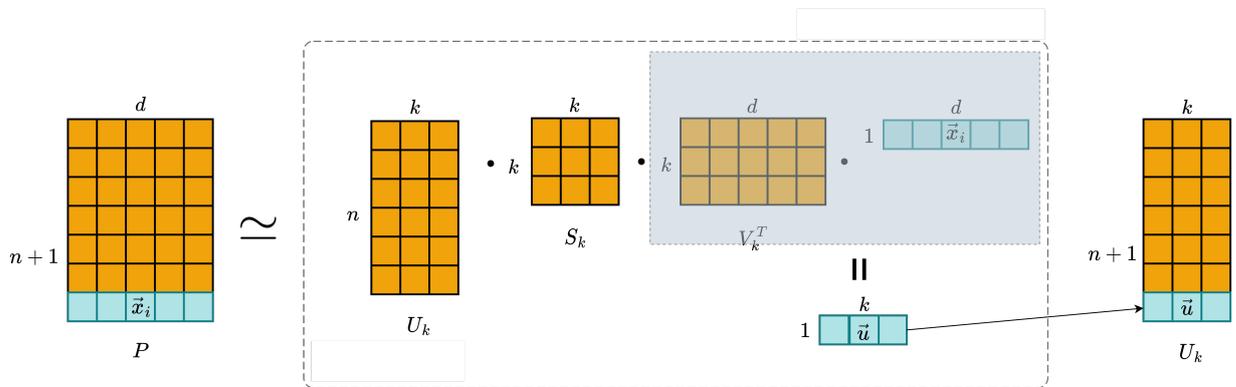


Figura 16 – Projeção de um novo indivíduo de alta dimensão  $x_i$  através da aproximação de baixa ordem  $\phi(x_i \in P)$

Fonte: (FONSECA *et al.*, 2022)

O melhor valor de  $k$  para minimizar a diferença entre  $P$  e  $\tilde{P}$  pode ser descrito

pelo lema de Johnson-Lindenstrauss (JL), adaptado para o contexto de otimização conforme descrito abaixo:

**Lema 5.1** (Johnson-Lindenstrauss). *Para qualquer  $0 < \epsilon < 1$  e inteiro  $n$ , seja  $k$  um inteiro positivo tal que  $k \geq 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \log n$ . Então, para qualquer conjunto  $P$  de  $n$  pontos em  $\mathbb{R}^d$ , existe uma função  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^k$  tal que para todos os  $x, y \in P$ ,*

$$(1 - \epsilon) |x - y|^2 \leq |\phi(x) - \phi(y)|^2 \leq (1 + \epsilon) |x - y|^2.$$

*Além disso, esta função pode ser encontrada em tempo polinomial.*

LSMDE utiliza o JL para inferir a melhor dimensionalidade  $k$  para a redução de dimensionalidade por Decomposição em Valores Singulares, a fim de preservar as distâncias entre indivíduos por um fator de  $(1 \pm \epsilon)$ , como apresentado na Figura 17. Observa-se que um aumento da distorção admissível  $\epsilon$  permite reduzir drasticamente o número mínimo de dimensões para um determinado número de indivíduos de uma população. Contudo, isso pode dificultar a precisão da otimização no espaço reduzido.

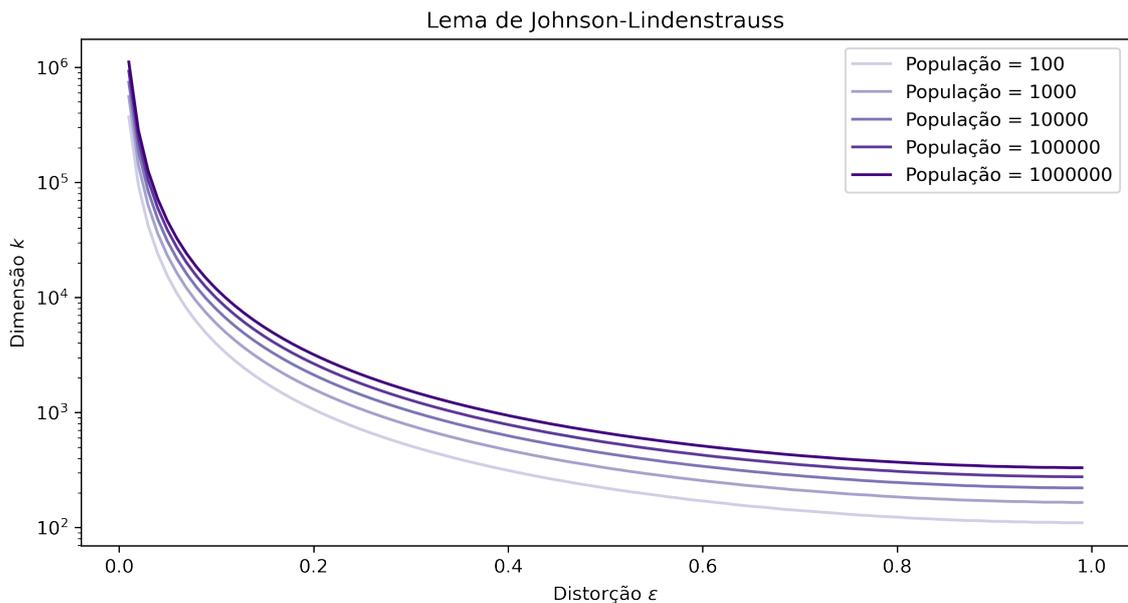


Figura 17 – Cálculo do lema de Johnson-Lindenstrauss para diferentes tamanhos de população do LSM-DE

Fonte: Autor

Ao não fazer suposições sobre a topologia do espaço de busca, o LSMDE é mais flexível e pode ser aplicado a diferentes espaços de busca. Com base no JL e na definição 4.3.1, existe uma função adaptativa  $\phi$  que mapeia cada elemento de alta dimensão  $x_i \in P$  em seu elemento correspondente  $\psi_i \in \Psi$  e uma função inversa  $\phi'$  que projeta esses indivíduos de baixa dimensão de volta para o espaço de alta dimensão, de modo que:

$$\phi(x_i) = \psi_i, \phi'(\psi_i) \simeq x_i \mid f(x_i) \simeq f(\phi'(\psi_i)), \forall i \in [1, n] \quad (68)$$

$$\phi'(\psi_i) = \psi_i \cdot V_k^T \quad (69)$$

onde  $f$  é a função de aptidão para o problema de otimização e  $V_k^T$  é uma matriz de reconstrução. A Figura 18 ilustra esse processo de transformação de espaço.

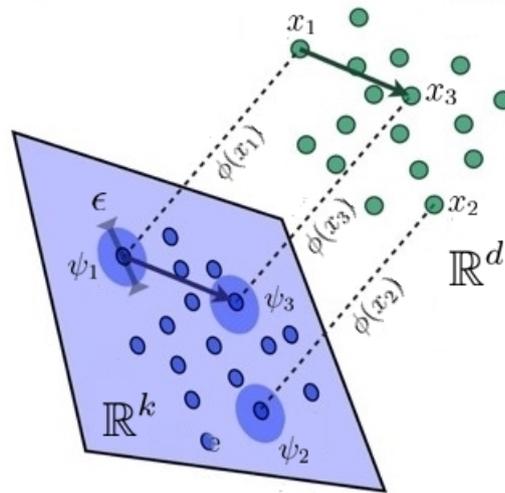


Figura 18 – Ilustração do processo de transformação de espaço  $\mathbb{R}^d \rightarrow \mathbb{R}^k$

Fonte: Autor

A partir dos conceitos apresentados acima, desenvolve-se uma função  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^k$  que mapeia soluções candidatas de um espaço de alta dimensão  $P$  para um espaço de baixa dimensionalidade  $\Psi$ . Da mesma forma, desenvolve-se uma função  $\phi' : \mathbb{R}^k \rightarrow \mathbb{R}^d$  de modo que cada indivíduo  $\psi \in (\Psi \subset \mathbb{R}^k)$  possa ser mapeado de volta para o indivíduo  $x \in (P \subset \mathbb{R}^d)$ , conforme os Algoritmos 2 e 3 a seguir:

---

**Algoritmo 2** SVD Adaptativo  $\phi$

---

**Entrada:**  $P, n, \epsilon$

**Saída:**  $\tilde{P}$

**início**

$k \leftarrow 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \log n ;$

$U_k, S_k, V_k \leftarrow SVD(P) ;$

$\Psi \leftarrow (U_k S_k) ;$

**Retorno:**  $\Psi, V_k^T ;$

**fim**

---



---

**Algoritmo 3** Inversa do SVD Adaptativo  $\phi'$

---

**Entrada:**  $\Psi, V_k^T$

**Saída:**  $\tilde{P}$

**início**

$\tilde{P} \leftarrow \Psi V_k^T ;$

**fim**

---

A complexidade computacional do LSMDE é limitada assintoticamente pela complexidade do processo de transformação de espaço definido nos Algoritmos 2 e 3, uma vez que essas transformações são responsáveis pelo maior custo computacional. De acordo com [Golub e Van Loan \(2013\)](#), a complexidade computacional do método SVD representado pelo Algoritmo 2, linha 2 é  $O(k^2n)$ . A complexidade da multiplicação de matrizes para gerar as representações de baixa dimensionalidade do espaço de busca  $\Psi$  também é  $O(k^2n)$ . No Algoritmo 3, linha 1, a complexidade da projeção de uma população de baixa dimensionalidade de volta ao espaço de alta dimensionalidade é  $O(knd)$ . Em resumo, a complexidade de converter uma população para baixa dimensionalidade e recuperar a população para alta dimensionalidade é  $O(k^2n + knd)$ .

Como motor do processo de otimização, LSMDE aplica um algoritmo de evolução diferencial denominado SHADE, hibridizado com um modelo de mistura gaussiana para descobrir regiões de optimalidade que podem ser convertidas de volta ao espaço original. Essa hibridização foi denominada GM-SHADE (*Gaussian Mixture SHADE*).

### 5.1.3 Otimização & Agrupamento em $\Psi$

Para guiar o processo de busca por regiões promissoras em um espaço de busca de baixa dimensionalidade, optou-se por utilizar um algoritmo de evolução diferencial denominado SHADE em conjunto com o *Clustering* por modelos de mistura gaussiana. Esta metaheurística foi escolhida devido sua ampla aplicabilidade em problemas LSGO e por possuir uma taxa de cruzamento (CR) e um fator de escala (F) atualizados dinamicamente seguindo as mesmas regras definidas em [Molina, LaTorre e Herrera \(2018\)](#), o que diminui o número de parâmetros configuráveis.

O GM-SHADE busca por regiões promissoras no espaço de baixa dimensionalidade  $\Psi$ , agrupando-as em *clusters* que são representados por uma função composta de várias gaussianas, cada uma identificada por  $\lambda \in 1, \dots, \Lambda$ , onde  $\Lambda$  é o número atual de *clusters*. A cobertura do *cluster* é determinada por uma gaussiana  $\lambda$ , composta por um indivíduo médio  $\mu$ , uma covariância  $\Sigma$  e uma probabilidade de mistura  $\pi$ . O objetivo é garantir que cada gaussiana se ajuste a todas as soluções candidatas  $\psi_i$  pertencentes a cada *cluster*.

O processo de assimilação trabalha como um classificador, conservando no sistema somente informações relevantes e direcionando a busca para regiões supostamente promissoras. O objetivo é reunir soluções similares no mesmo *cluster*, mantendo uma solução no centro do *cluster* que seja representativa para as demais soluções. Para evitar um esforço computacional extra, o agrupamento é desenvolvido como um processo iterativo, no qual os *clusters* são progressivamente alimentados por novas soluções geradas pelo algoritmo SHADE.

O processo de assimilação ao *cluster* envolve a atribuição de uma observação específica, neste caso  $\psi_i$ , ao *cluster*  $\lambda$  que mais provavelmente teria gerado essa

observação. Este procedimento é especialmente relevante em modelos de misturas gaussianas (GMMs), onde a ideia é modelar a distribuição de um conjunto de dados como uma mistura de várias distribuições gaussianas.

O primeiro passo para a assimilação é calcular a probabilidade a posteriori que cada *cluster* tem pela observação  $\psi_i$ . Matematicamente,  $\gamma(z_{\lambda i})$  é definido pela seguinte fórmula:

$$\gamma(z_{\lambda i}) = \frac{\pi_{\lambda} \cdot \mathcal{N}(\psi_i | \mu_{\lambda}, \Sigma_{\lambda})}{\sum_{k=1}^K \pi_k \cdot \mathcal{N}(\psi_i | \mu_k, \Sigma_k)}$$

Aqui,  $\pi_{\lambda}$  é o peso do *cluster*  $\lambda$ , e  $\mathcal{N}(\psi_i | \mu_{\lambda}, \Sigma_{\lambda})$  é a função densidade de probabilidade da distribuição gaussiana associada ao *cluster*  $\lambda$ , avaliada em  $\psi_i$ .

Uma vez calculadas as probabilidades a posteriori para todos os *clusters*, o LSMDE determina qual *cluster* tem a maior probabilidade sobre a observação  $\psi_i$ . Isso é feito através da maximização da probabilidade a posteriori:

$$\lambda^* = \arg \max_{\lambda} \gamma(z_{\lambda i})$$

O *cluster*  $\lambda^*$  é, então, aquele ao qual a observação  $\psi_i$  é assimilada. Em outras palavras, sob a suposição do modelo GMM,  $\psi_i$  é mais provável de ter sido gerado pelo *cluster*  $\lambda^*$ .

O conceito de probabilidade a posteriori reflete a probabilidade condicional de um indivíduo  $\psi_i$  pertencer a um *cluster*  $\lambda$ , dadas as propriedades estatísticas desse *cluster*. Assim, a assimilação não é um processo de classificação "CRISP", mas sim uma forma de classificação "SOFT". Isso significa que cada indivíduo tem uma probabilidade de pertencer a cada *cluster*, mas é finalmente atribuído ao *cluster* para o qual essa probabilidade é máxima.

Em resumo, o processo de assimilação ao *cluster* permite que cada observação  $\psi_i$  seja efetivamente atribuída a um dos *clusters*, maximizando a probabilidade condicional sob o modelo estatístico adotado.

Uma das tarefas-chave na aplicação de modelos de mistura é a determinação de um número adequado de *clusters*. O LSMDE detecta automaticamente esse número usando um parâmetro baseado em um modelo de mistura finito com processo de Dirichlet (BLEI; JORDAN, 2006). O processo de Dirichlet fornece uma estrutura bayesiana não paramétrica para descrever distribuições sobre modelos de mistura com um número infinito de componentes de mistura (*clusters*). Um processo de Dirichlet (DP) pode ser descrito a partir de seus parâmetros iniciais:

- **Medida de Base ( $G_0$ ):** Uma gaussiana é frequentemente escolhida como a medida de base, caracterizada por parâmetros  $\mu$  e  $\Sigma$  da distribuição gaussiana com base nos dados.

- **Parâmetro de Concentração ( $\omega$ ):** Este hiperparâmetro controla a probabilidade de formação de um novo *cluster*. Valores maiores de  $\omega$  facilitam a criação de mais *clusters*.

A partir disso, o Agrupamento por modelo de mistura gaussiana com DP pode ser descrito como:

1. **Inicialização:** Todos os indivíduos,  $\psi_i$ , podem inicialmente ser atribuídos a um único *cluster*.
2. **Iteração:** Para cada indivíduo  $\psi_i$ :
  - Remova  $\psi_i$  do *cluster* atual e atualize os parâmetros do mesmo.
  - Calcule a probabilidade de  $\psi_i$  pertencer a cada *cluster* existente utilizando:

$$\gamma(z_{\lambda i}) = \frac{\pi_{\lambda} \cdot \mathcal{N}(\psi_i | \mu_{\lambda}, \Sigma_{\lambda})}{\sum_{k=1}^K \pi_k \cdot \mathcal{N}(\psi_i | \mu_k, \Sigma_k)}$$

- Calcule a probabilidade de  $\psi_i$  iniciar um novo *cluster*.
  - Reatribua  $\psi_i$  a um *cluster* existente ou a um novo *cluster*, maximizando  $\gamma(z_{\lambda i})$ .
3. **Convergência:** O algoritmo converge quando as atribuições dos *clusters* se estabilizam ou após um número predeterminado de iterações.

Em resumo, o processo de Dirichlet fornece um método robusto e flexível para determinar o número de *clusters* de forma adaptativa. Em cada iteração, cada indivíduo é removido temporariamente de seu *cluster* atual, e a probabilidade desse indivíduo pertencer a cada um dos *clusters* existentes é recalculada. Com base nesses cálculos de probabilidade, o indivíduo é reatribuído. Ele pode ser reatribuído ao seu *cluster* original, a um diferente *cluster* já existente ou até mesmo formar um novo *cluster*. Se um novo *cluster* é formado, então o número total de *clusters* aumenta. Se um *cluster* existente fica vazio após esta reatribuição, esse *cluster* será efetivamente removido, e o número total de *clusters* será reduzido. Em suma, os modelos de mistura de processo de Dirichlet podem ser implementados como um modelo de mistura gaussiano (GMM) no qual todos os parâmetros, incluindo  $\Lambda$ , são inferidos a partir de  $\Psi$ . No GMM, gaussianas com valores elevados de  $\pi$  representam regiões onde mais indivíduos  $\psi$  são observados e são consideradas regiões promissoras. A Figura 19 mostra o número de *clusters* sendo atualizado de acordo com a população  $\Psi$ .

Nesta tese, considera-se que uma gaussiana  $\lambda$  é promissora se:

$$\pi_{\lambda} \geq 0.25(1/\Lambda) + (1/\Lambda) \quad (70)$$

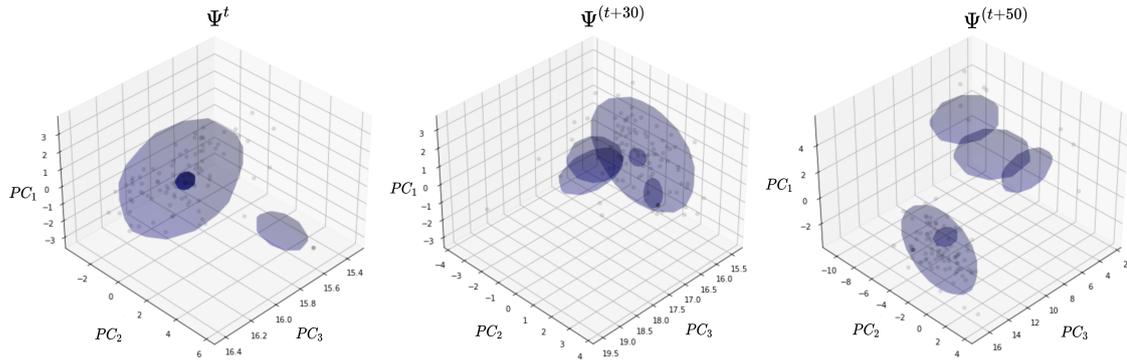


Figura 19 – Modelos de mistura gaussiana encontrando regiões promissoras no espaço de busca de baixa dimensionalidade da função Alpine N.1. O número de *clusters* é automaticamente inferido. Neste caso, o número de *clusters* inicialmente definido (população  $\Psi^t$ ) não corresponde à verdadeira distribuição geradora das próximas populações; portanto, o modelo de mistura gaussiana ajusta o número de *clusters* de acordo com a distribuição atual da população.

Fonte: (FONSECA *et al.*, 2022)

onde  $\Lambda$  é o número atual de *clusters* inferido a partir do processo de Dirichlet para a população atual  $\Psi$ . A gaussiana promissora definida por  $\theta = [\pi_p, \mu_p, \Sigma_p]$  representa a região do espaço de busca onde há maiores chances de encontrar soluções significativas para o problema e  $\mu_p$  é o indivíduo médio que representa a gaussiana mais promissora da iteração atual. Em cada geração, um vetor mutante  $v_i$  é gerado a partir de um membro  $\psi_i$  da população de baixa dimensionalidade atual de acordo com a Equação a seguir:

$$v_i = \psi_i + F_i \cdot (\mu_p - \psi_i) + F_i \cdot (\psi_{r1} - \psi_{r2}) \quad (71)$$

$\mu_p$  é o indivíduo médio da gaussiana mais promissora na geração atual, e o fator de escala  $F_i$  é gerado a partir da distribuição de Cauchy, conforme definido em Molina, LaTorre e Herrera (2018). Os índices  $r1$  e  $r2$  indicam que os indivíduos  $\psi_{r1}$  e  $\psi_{r2}$  são escolhidos aleatoriamente da população  $\Psi$ . Essa estratégia foi denominada *DE/promising/2*, uma vez que utiliza o centro do *cluster* promissor como base da mutação e usa dois pares de vetores para compor o vetor mutante.

Após gerar o vetor mutante  $v_i$ , ele é cruzado com o pai  $\psi_i$  para gerar o vetor de teste  $u_i$  da seguinte forma:

$$u_{i,j} = \begin{cases} v_{i,j} & , \text{ se } \text{rand}[0,1) \leq CR_i \\ \psi_{i,j} & , \text{ caso contrário} \end{cases} \quad (72)$$

$\text{rand}[0,1)$  denota um número aleatório uniformemente selecionado em  $[0,1)$ ,  $u_{i,j}$  é a  $j$ -ésima dimensão do vetor de teste  $u$  do  $i$ -ésimo indivíduo e  $CR_i \in [0,1]$  é a taxa de cruzamento do indivíduo  $\psi_i$ .

Após todos os vetores de teste  $u_i$  serem gerados, um processo de seleção determina os sobreviventes para a próxima população de baixa dimensionalidade  $\Psi^{(t+1)}$ . O

operador de seleção compara cada  $\psi_i$  com seu correspondente vetor de teste  $u_i$ , mantendo o melhor vetor na população. Em um problema de minimização, o cruzamento é definido da seguinte forma:

$$\psi_i = \begin{cases} u_i & , \text{ se } f(\phi'(u_i)) \leq f(\phi'(\psi_i)) \\ \psi_i & , \text{ caso contrário} \end{cases} \quad (73)$$

Em que  $f(\phi)'$  é o valor da aptidão da inversa da função de transformação de espaço, ou seja, o valor de aptidão do indivíduo de baixa dimensionalidade. Quando uma condição de parada é alcançada, o GM-SHADE retorna  $\theta = [\pi_p, \mu_p, \Sigma_p]$  da gaussiana promissora da população atual  $\Psi$ . Em um problema multimodal, mais de um gaussiana podem ser promissoras. A Figura 19 ilustra o agrupamento do espaço de busca reduzido à medida que a população  $\Psi$  é atualizada em cada geração. O indivíduo médio  $\mu_p$  é retornado para o espaço de busca original usando a função SVD adaptativa inversa  $\phi'$  da seguinte forma:

$$\phi'(\mu_p) = \mu_p \cdot V_k^T \quad (74)$$

A gaussiana promissora com parâmetro  $\theta = [\pi_p, \mu_p, \Sigma_p]$  reconvertida em uma vizinhança promissora de alta dimensionalidade pode ser explorada por um algoritmo de busca local especializado com o objetivo de encontrar a solução ótima para o problema LSGO em uma região limitada do espaço de busca original aproximado (Figura 20).

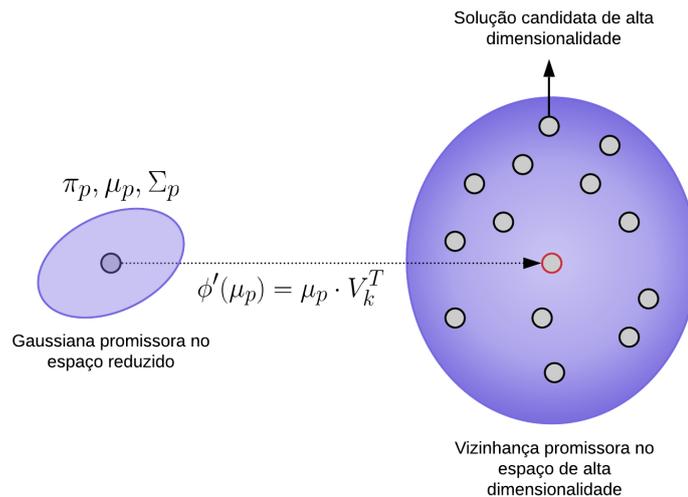


Figura 20 – Ilustração de uma transformação de espaço de uma Gaussiana promissora em  $\Psi$  para uma região promissora em  $\tilde{P}$ .

Fonte: (FONSECA *et al.*, 2022)

### 5.1.4 Busca Local

Uma combinação de duas abordagens foi utilizada para explorar a vizinhança promissora de alta dimensionalidade: MTS-LS1 (TSENG; CHEN, 2008), especialmente projetado para LSGO e L-BFGS-B (MORALES; NOCEDAL, 2011) que usa uma aproximação do gradiente.

O MTS começa gerando  $n$  soluções promissoras usando a inversa da transformação de espaço sobre a gaussiana promissora  $\theta$  como ilustrado na Figura 21

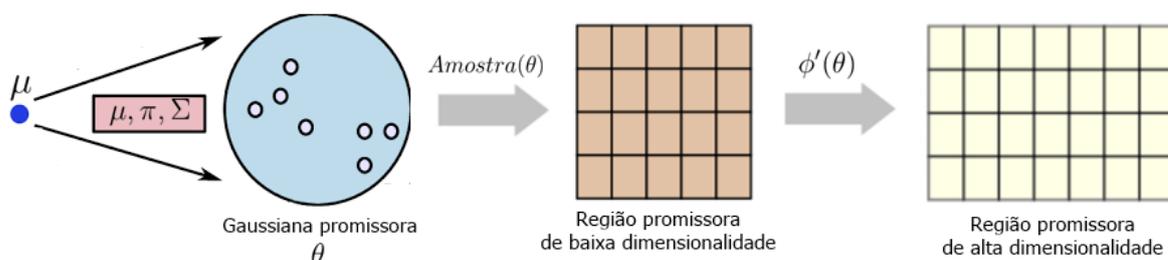


Figura 21 – A gaussiana promissora  $\theta$  representada pelo seu centro  $\mu$  é capaz de gerar uma amostra de  $n$  soluções promissoras de baixa dimensionalidade. Essas soluções podem ser transformadas para o espaço de original de alta dimensionalidade através da função de transformação  $\phi'$ .

Fonte: Autor

O MTS consiste em uma série de iterações de busca que terminarão quando o número máximo de avaliações da função objetivo for atingido. O MTS executa buscas nas soluções iniciais promissoras apenas na primeira iteração. Mas nas iterações seguintes, apenas algumas das melhores soluções são selecionadas como soluções de primeiro plano e o MTS executa buscas sobre essas soluções. Depois de executar a busca por soluções em primeiro plano, a MTS aplica uma busca local 1 (LS1) na melhor solução atual com o objetivo de melhorar a melhor solução até o momento. No final de uma iteração, melhores soluções são selecionadas como soluções de primeiro plano para próxima iteração.

A busca local selecionada é o L-BFGS-B, um tipo de algoritmo de otimização de segunda ordem e pertencente a uma classe de métodos Quasi-Newton (que aproximam a segunda derivada para os problemas onde não pode ser calculada diretamente). O método de Newton usa a matriz Hessiana. No entanto, tem uma limitação, pois requer o cálculo da matriz inversa Hessiana que pode ser computacionalmente intensivo. O método Quasi-Newton aproxima a matriz inversa Hessiana usando o gradiente e, portanto, pode ser computacionalmente viável. O método L-BFGS-B atualiza o cálculo da matriz Hessiana a cada iteração, em vez de recalculá-la. No entanto, o tamanho da Hessiana ( $H$ ) e da sua inversa ( $\bar{H}$ ) depende do número de parâmetros de entrada da função objetivo. Portanto, para um problema LSGO, o L-BFGS transpõe essa dificuldade assumindo uma simplificação da matriz inversa Hessiana na iteração anterior. Ao

contrário do BFGS, que é baseado no histórico completo dos gradientes, o L-BFGS-B é baseado nos  $m$  gradientes mais recentes.

O algoritmo 4 apresenta o método LSMDE a partir da interação entre as etapas anteriores. Detalhes de implementação foram suprimidos para simplificar o entendimento.

**Algoritmo 4** LSMDE**Entrada:**  $d, n, \epsilon, \omega, CR, F$ **Saída:**  $\mu$ **início** $P \leftarrow rand(n)$  ; $opt \leftarrow MTS(P)$  ;**repita****para** cada  $x_i \in P$  **faça** $\hat{x}_i \leftarrow OBL(x_i)$  ; $x_i = \min(f(x_i), f(\hat{x}_i))$  ;**fim****se**  $U_k, S_k, V_k$  **existem** **então****para**  $x_i \in P$  **faça** $r_i \leftarrow x_i V_k S_k^{-1}$  ; $U_k \leftarrow U_k \cup r_i$  ;**fim****fim****else** $U, S, V^T \leftarrow SVD(P)$  ; $k \leftarrow 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \log n$  ; $U_k \leftarrow U[:, :k]$  ; $S_k \leftarrow diag(S[:, :k])$  ; $V^T \leftarrow V[:, :k]$  ;**end** $\Psi \leftarrow U_k S_k$  ;**repita** $\pi, \mu, \Sigma \leftarrow GMM(\Psi, \omega)$  ;**para** cada  $\psi \in \Psi$  **faça** $v_i = \psi_i + F_i \cdot (\mu_p - \psi_i) + F_i \cdot (\psi_{r1} - \psi_{r2})$  ; $u_i = \begin{cases} v_i & , \text{ se } rand[0,1] \leq CR_i ; \\ \psi_i & , \text{ caso contrário} \end{cases}$  ; $\psi_i = \begin{cases} u_i & , \text{ se } f(\phi'(u_i)) \leq f(\phi'(\psi_i)) ; \\ \psi_i & , \text{ caso contrário} \end{cases}$  ;**fim**Atualize  $CR$  e  $F$  ;**até** algum  $\pi_\lambda > \rho$  ; $P \leftarrow \phi'(\mu_p)$  ; $x^* \leftarrow MTS(P)$  ;**se**  $x^* \leq opt$  **então** $opt \leftarrow x^*$  ;**fim****até** condição de parada;**Retorno:**  $opt$  ;**fim**

## 5.2 PROCEDIMENTOS METODOLÓGICOS

De acordo com a Literatura, o *benchmark* para o *IEEE CEC 2008 Special Session and Competition on Large Scale Global Optimization* foi a primeira tentativa de propor um conjunto de problemas de otimização com o objetivo de fornecer uma plataforma de avaliação adequada para testar e comparar algoritmos de otimização em larga escala (TANG; YÁO *et al.*, 2007). Esse *benchmark* continha 3 funções separáveis (*Sphere*, *Rastrigin* e *Ackley*) e 4 funções totalmente não-separáveis (*Schwefel*, *FastFractal DoubleDip*, *Rosenbrock* e *Griewank*). As instancias de 100, 500 e 1000 dimensões dessas funções foram implementadas em C, Java e Matlab. O número máximo de avaliações da função objetivo foi fixado em  $5000 * D$ , em que  $D$  é a dimensionalidade do problema.

O *benchmark CEC'2008* foi bem-sucedido na construção de um conjunto escalonável de funções para promover pesquisas no campo da otimização global em larga escala. No entanto, o conjunto apresentava algumas deficiências significativas. Em primeiro lugar, o número de funções neste conjunto era limitado (apenas sete problemas). Portanto, era difícil comparar estatisticamente três ou mais métodos de otimização usando esse número limitado de instâncias. Em segundo lugar, os problemas incluídos eram totalmente separáveis (sem interação entre qualquer par de variáveis) ou totalmente não-separáveis (com uma forte interação entre qualquer par de variáveis possível). Como resultado, O *CEC'2008* foi expandido para incluir mais problemas com propriedades mais diversas.

Funções parcialmente separáveis foram incluídas pela primeira vez no *benchmark CEC'2010* (TANG; LI *et al.*, 2010). O *CEC'2010* é formado por 20 funções de teste, cada uma com 1000 dimensões. Pela primeira vez, uma nova categoria de problemas denominada funções parcialmente separáveis foi incluída no conjunto. Por definição, um problema parcialmente separável é uma função em que um número  $m$  de seus parâmetros não são independentes. Todos os 20 problemas foram criados com base em seis funções básicas. O limite de avaliações da função objetivo também foi diminuído para 3 milhões, ou seja, 2 milhões a menos que o anteriormente proposto no *CEC'2008* (o que torna a otimização mais difícil). Como resultado, resolver o novo conjunto era mais desafiador do que seu antecessor.

Proposto no *IEEE CEC 2013 Special Session and Competition on Large Scale Global Optimization*, o *CEC'2013* foi desenvolvido especialmente para melhor representar as características de uma ampla gama de problemas do mundo real, bem como colocar novos desafios para os algoritmos existentes, especialmente para algoritmos baseados em decomposição (LI, X. *et al.*, 2013). Os novos tipos de problemas que foram incluídos neste *benchmark* são:

- problemas com tamanhos de componentes não uniformes

- problemas com coeficientes de componentes desiguais
- problemas com subproblemas sobrepostos

Os primeiros dois tipos de problemas representam cenários em que alguns componentes de uma determinada função são mais difíceis de resolver (ou seja, possuem um custo computacional maior) ou têm uma influência mais substancial no desempenho da solução. A categoria de problemas sobrepostos é um caso particular de problemas não separáveis onde algumas variáveis têm interação com algumas outras variáveis e onde não há uma única variável ou grupo de variáveis que podem ser identificados como um componente isolado. Portanto, esses problemas não podem ser categorizados como funções parcialmente separáveis.

Com o objetivo de fornecer um conjunto de referência conveniente e flexível que permita comparar o desempenho do LSMDE, esta tese adota o *benchmark CEC'2013* para a análise de performance. Optou-se por usar este conjunto de funções por ser o mais recente e uma extensão dos *benchmarks* anteriores. O *CEC'2013* se distingue de seus predecessores por incorporar uma gama mais ampla de problemas de otimização em larga escala que se assemelham a problemas do mundo real, tornando-os mais desafiadores de otimizar. O *CEC'2013* é composto por 15 funções de teste com 1000 dimensões (em geral, essas funções podem ser categorizadas como mostrado na Tabela 4). O limite de avaliações das funções objetivo foram divididas em 3 grupos de experimentos:  $1,2 \times 10^5$ ,  $6 \times 10^5$  e  $3 \times 10^6$ . O mínimo de todas as funções propostas nos problemas no *CEC'2013* é zero, com o exceção das funções sobrepostas, para as quais ainda não se sabe como a inclusão de variáveis de decisão compartilhadas pode alterar o valor mínimo. Para comparar vários algoritmos usando este conjunto de funções de *benchmark*, cada algoritmo recebe um limite de avaliações fixo. Para comparar vários algoritmos, essa tese utilizou o método de análise de variância não paramétrico de Kruskal-Wallis para detectar diferenças adotando um nível de significância  $\alpha = 0,05$ . Se uma diferença significativa for detectada e a fim de considerar a taxa de erro para múltiplas comparações, essa tese utilizou dois procedimentos post-hoc, Conover e Dunn, com correção de Holm para os ajustes do p-valor, como sugerido em [Ostertagova, Ostertag e Kováč \(2014\)](#).

Esta tese observou que as análises realizadas sob o *CEC'2013* cobrem apenas uma parte das métricas de performance interessantes para problemas LSGO, uma vez que são organizadas apenas em uma única dimensão  $1000 - D$  (embora os problemas, em princípio, sejam escaláveis) e a avaliação de desempenho é prescrita para apenas alguns limites específicos ( $1,2 \times 10^5$ ,  $6 \times 10^5$  e  $3 \times 10^6$  avaliações). Essa configuração não permite medir de forma confiável o comportamento da escalabilidade com a dimensão, uma das características mais importantes que um experimento de *benchmarking* para algoritmos LSGO deve investigar.

Outro ponto de atenção é que o *CEC'2013* não permite investigar se as dificuldades dos pontos de otimização-alvo são comparáveis entre os problemas e dimensões; no entanto, essa semelhança é necessária para agregar adequadamente os desempenhos em diferentes problemas e investigar o comportamento de escalabilidade com a dimensão do problema. Para cobrir os pontos apresentados, além da comparação do método proposto utilizando o *CEC'2013*, esta tese amplia os experimentos com uma análise de escalabilidade adicional utilizando a suíte de teste *bbob-largescale*.

A suíte de teste *bbob-largescale* oferece 24 funções em seis dimensões (20, 40, 80, 160, 320, 640) dentro do *framework* COCO e traz, além de espaços de busca já cobertos pelo *CEC'2013*, uma nova variedade de características de espaços, tais como: funções com mal, baixo ou moderado condicionamento, funções multimodais com adequada e fraca estrutura global, entre outros (mais detalhes sobre essas funções podem ser encontrados em [Elhara et al. \(2019\)](#)). Todas as 24 funções são, a princípio, escaláveis para uma dimensão arbitrária, mas por medida de comparação, esta tese utilizou as dimensões sugeridas pelo *framework* COCO. A suíte para larga escala é derivada da suíte de teste *bbob* existente, que é de único objetivo e sem restrições.

Esta tese aprofunda os aspectos técnicos dos experimentos no *CEC'2013* e *bbob-largescale* nas subsecções seguintes a partir de duas perspectivas: análise de performance e análise de escalabilidade.

### 5.2.1 Análise de performance

Para avaliar o desempenho do método LSMDE proposto, adotou-se o conjunto de testes mais recente para problemas LSGO realizado em 2013 no *Congresso de Computação Evolutiva do IEEE (CEC'2013)* ([LI, X. et al., 2013](#)), utilizando novos critérios de avaliação realizados em 2019 no *Congresso de Computação Evolutiva do IEEE (CEC'2019)* ([MOLINA; LATORRE, 2018](#)). O conjunto de testes *CEC'2013* é composto por 15 funções de minimização com 1000 dimensões, exceto  $f_{14}$  e  $f_{15}$ , que são funções sobrepostas com  $d = 905$ . As funções são divididas em 5 categorias: funções totalmente separáveis ( $f_1 - f_3$ ), funções com subcomponentes separáveis ( $f_4 - f_7$ ), funções sem subcomponentes separáveis ( $f_8 - f_{11}$ ), funções sobrepostas (ou com sobreposição) ( $f_{12} - f_{14}$ ) e funções não-separáveis ( $f_{15}$ ). Além disso, o conjunto de testes *CEC'2013* melhorou versões anteriores ao incluir tamanhos de subcomponentes não uniformes, desequilíbrio na contribuição dos subcomponentes, funções com subcomponentes sobrepostos, novas transformações como quebra de simetria, mal-condicionamento e irregularidades locais ([OMIDVAR; LI; YAO, 2021b](#); [LI, X. et al., 2013](#)).

O *CEC'2013* foi projetado para fornecer uma avaliação adequada dos algoritmos de otimização em larga escala ([LI, X. et al., 2013](#)). Com esse objetivo, suas funções representam com sucesso a natureza de uma variedade de problemas do mundo real

e constroem um conjunto escalável de problemas para promover pesquisas no campo de LSGO (OMIDVAR; LI; TANG, 2015). A Tabela 1 apresenta algumas características do CEC'2013.

Tabela 1 – Resumo das funções de *benchmark* CEC'2013

	Função	Propriedades	intervalo de busca
$f_1$	Elliptic Function	Unimodal	$[-100, 100]^d$
$f_2$	Rastrigin Function	Multimodal	$[-5, 5]^d$
$f_3$	Ackley Function	Multimodal	$[-32, 32]^d$
$f_4$	Elliptic Function	Unimodal	$[-100, 100]^d$
$f_5$	Rastrigin Function	Multimodal	$[-5, 5]^d$
$f_6$	Ackley Function	Multimodal	$[-32, 32]^d$
$f_7$	Schwefels Problem 1,2	Multimodal	$[-100, 100]^d$
$f_8$	Elliptic Function	Unimodal	$[-100, 100]^d$
$f_9$	Rastrigin Function	Multimodal	$[-5, 5]^d$
$f_{10}$	Ackley Function	Multimodal	$[-32, 32]^d$
$f_{11}$	Schwefels Problem 1,2	Unimodal	$[-100, 100]^d$
$f_{12}$	Rosenbrock's Function	Multimodal	$[-100, 100]^d$
$f_{13}$	Schwefels Function	Unimodal	$[-100, 100]^d$
$f_{14}$	Schwefels Function	Unimodal	$[-100, 100]^d$
$f_{15}$	Schwefels Problem 1,2	Unimodal	$[-100, 100]^d$

Fonte: Autor

Inicialmente, sete abordagens metaheurísticas de ponta são selecionadas para participar da comparação com LSMDE: BICCA (GE; ZHAO *et al.*, 2020), CC-CMA-ES (LIU; TANG, 2013), C-DEEPSO (MARCELINO *et al.*, 2018), GL-SHADE (PACHECO-DEL-MORAL; COELLO, 2020), IHDELS (MOLINA; HERRERA, 2015), MLSHADE-SPA (HADI; MOHAMED; JAMBI, 2019) e SHADE-ILS (MOLINA; LATORRE; HERRERA, 2018). Em seguida, uma segunda rodada de comparações é feita considerando os melhores algoritmos do teste anterior e os algoritmos que possuem características similares a proposta de tese: LS-EDA (DONG; WANG; ZHOU, 2019), MSES (FENG *et al.*, 2021) e OJS (RODRIGUES *et al.*, 2023). As metaheurísticas concorrentes foram escolhidas de acordo com suas classificações na *IEEE CEC Special Sessions and Competitions on Large-Scale Global Optimization* e os critérios apresentados no capítulo 2 de revisão da literatura.

Com base nos critérios definidos em Molina e LaTorre (2018), foram realizados três experimentos com diferentes números máximos de avaliações da função objetivo (FE):  $1,2 \times 10^5$  avaliações,  $6 \times 10^5$  avaliações e  $3 \times 10^6$  avaliações. 25 execuções independentes por função foram realizadas para cada experimento e algoritmo. As médias dos resultados de cada metaheurística concorrente usada neste tese são descritas em trabalhos recentes, reunidos nas *Special Sessions and Competition on Large-Scale Global Optimization* e disponíveis em TFSLGO. Os parâmetros de controle

adotados pelos algoritmos concorrentes estão descritos em seus respectivos artigos devidamente referenciados nesta tese, e os parâmetros de controle adotados pelo LSMDE são mostrados na Tabela 2. A tolerância  $\epsilon$  especifica a distorção permitida ao construir o espaço de baixa dimensionalidade  $\Psi$ . Valores mais altos de  $\epsilon$  indicam uma maior tolerância para a distorção e menores valores de  $k$ . No entanto, à medida que a perda de informação aumenta, o procedimento de otimização se torna mais difícil.  $\omega$  representa o parâmetro de concentração do processo de Dirichlet. Intuitivamente, o processo de Dirichlet tem a mesma probabilidade de iniciar um novo agrupamento para uma solução candidata  $\psi_i$  quanto de adicionar essa solução a um agrupamento com  $\omega$  elementos. Um valor  $\omega$  maior significa mais agrupamentos. O tamanho da população foi escolhido de forma a tentar garantir que os algoritmos tivessem as mesmas condições iniciais de exploração, e  $\epsilon$  e  $\omega$  foram determinados experimentalmente.

Tabela 2 – Parâmetros usados no LSMDE

Parâmetro	Valor	Descrição
$n$	100	Tamanho da população
$d$	1000	Dimensão do problema
$\epsilon$	0.3	Taxa de tolerancia Johnson-Lindenstraus
$\omega$	1	Parâmetro de concentração do processo de Dirichlet

Fonte: Autor

A comparação de desempenho é realizada utilizando um método indicado pela *IEEE Task Force on Large-Scale Global Optimization* baseado na corrida de carros de Fórmula 1 (critério *Formula One*). Em particular, esse método foi usado na recente competição CEC LSGO 2019 (MOLINA; NESTERENKO; LATORRE, 2019). Esse processo de otimização de uma função é análogo a uma corrida, de modo que o competidor (algoritmo) que chega em primeiro lugar (melhor desempenho médio) recebe 25 pontos, o segundo lugar recebe 18 pontos, o terceiro recebe 15 pontos, e assim por diante. A Tabela 3 apresenta o resumo dos pontos obtidos na competição com oito competidores.

Tabela 3 – Resumo de pontuação

Posição	Pontuação
1º	25
2º	18
3º	15
4º	12
5º	10
6º	8
7º	6
8º	4

A Tabela 4 destaca a pontuação máxima que um competidor pode alcançar com base no número de funções contidas em cada uma das categorias de funções, permitindo uma análise mais profunda do desempenho dos algoritmos para cada categoria. Embora a pontuação máxima em um experimento seja de 375, como o número de funções por categoria não é igual, as pontuações máximas por categoria também não são iguais.

Tabela 4 – Resumo da pontuação máxima alcançável por categoria de função

Categoria	nº de funções	Pontuação máxima
Funções totalmente separáveis	3	75
Funções com subcomponentes separáveis	4	100
Funções sem subcomponentes separáveis	4	100
Funções com sobreposição	3	75
Funções não-separáveis	1	25

Fonte: Autor

### 5.2.2 Análise de escalabilidade

Como mencionado anteriormente, apenas a análise de desempenho do LSMDE em problemas de alta dimensionalidade não é o suficiente para mostrar a robustez do método ao aumento de dimensão. O segundo grupo de experimentos visa abordar essa característica em particular. Experimentos foram realizados na suíte *bbob-largescale*, que inclui 24 funções de objetivo único no domínio contínuo, com diferentes características: baixo ou moderado condicionamento, separabilidade e não separabilidade, multimodalidade, estrutura global fraca e outras. Uma visão geral dessas funções pode ser vista na Tabela 5 e fórmulas e descrições detalhadas podem ser encontradas em COCO. Os parâmetros usados pelo LSMDE são os mesmos descritos na Tabela 2.

O desempenho do LSMDE em termos de escalabilidade foi comparado com os seus dois melhores concorrentes: SHADE-ILS e GL-SHADE. Os resultados são apresentados considerando Funções de Distribuição Cumulativa Empírica (do inglês *empirical cumulative distribution function - ECDF*) também denominadas gráficos de distribuição de tempo de execução, fornecidas pelo framework COCO como padrão para novas suítes de teste.

Os gráficos de distribuição cumulativa empírica do COCO possuem 3 características principais:

- os valores-alvo não dependem das funções;
- os resultados para múltiplos valores-alvo são agregados em um único gráfico de distribuição;
- tentativas malsucedidas são calculadas através de simulações.

Tabela 5 – Resumo da suite de teste bbob-largescale

	Função	Propriedades
$f_1$	Sphere Function	Funções separáveis
$f_2$	Ellipsoidal Function	
$f_3$	Rastrigin Function	
$f_4$	Bueche-Rastrigin Function	
$f_5$	Linear Slope	
$f_6$	Attractive Sector Function	Funções com baixo ou moderado condicionamento
$f_7$	Step Ellipsoidal Function	
$f_8$	Rosenbrock Function, original	
$f_9$	Rosenbrock Function, rotated	
$f_{10}$	Ellipsoidal Function	Funções com alto condicionamento e unimodais
$f_{11}$	Discus Function	
$f_{12}$	Bent Cigar Function	
$f_{13}$	Sharp Ridge Function	
$f_{14}$	Different Powers Function	
$f_{15}$	Rastrigin Function	
$f_{16}$	Weierstrass Function	
$f_{17}$	Schaffers F7 Function	
$f_{18}$	Schaffers F7 Function, Funções moderadamente mal condicionadas	
$f_{19}$	Composite Griewank-Rosenbrock Function F8F2	
$f_{20}$	Schwefel Function	Funções multimodais com estrutura global fraca
$f_{21}$	Gallagher's Gaussian 101-me Peaks Function	
$f_{22}$	Gallagher's Gaussian 21-hi Peaks Function	
$f_{23}$	Katsuura Function	
$f_{24}$	Lunacek bi-Rastrigin Function	

Fonte: Autor

ECDF mostra a taxa de sucesso no eixo y, ou seja, a proporção de problemas resolvidos, para qualquer *budget* dado no eixo x (medido em número de avaliações de função dividido pela dimensão).

Um ECDF não corresponde a um único teste, a agregação é realizada em execuções com reinicializações independentes e em várias instâncias de uma função (Figura 28) ou grupos de funções (Figura 30). Se a taxa de sucesso em um determinado problema for menor que 1, mas maior que zero, o tempo de execução das tentativas malsucedidas é determinado por reinicializações simuladas a partir dos dados registrados de todas as tentativas no mesmo problema. As distribuições cumulativas empíricas permitem uma comparação quantificada entre os algoritmos: um deslocamento horizontal no gráfico corresponde a uma diferença de tempo de execução com o respectivo fator.

Seguindo o padrão estabelecido pelo COCO, esta tese manteve os 51 valores-alvo escolhidos uniformemente em escala logarítmica entre 100 e  $10^{-8}$ . Todos os ex-

perimentos foram executados em 24 funções. ECDF's mostram as porcentagens de valores-alvo alcançados com um determinado *budget* (mostrados no eixo y em escala logarítmica). Os valores-alvo são definidos como certas distâncias do valor ótimo.

## 6 RESULTADOS

### 6.1 ANÁLISE DE PERFORMANCE DO LSMDE COMPARADO COM O ESTADO DA ARTE

A Figura 22 apresenta a pontuação cumulativa média dos algoritmos concorrentes para os três experimentos propostos pelo *IEEE Task Force on Large-Scale Global Optimization* utilizando o critério Formula 1. Pode-se observar que o LSMDE tem a melhor pontuação média geral, o que indica uma possível superioridade em relação aos outros algoritmos. No entanto, é essencial observar o desempenho de cada algoritmo em cada categoria de função. Essa análise permite identificar em quais tipos de problemas LSGO os algoritmos se saíram melhor e pior. Para realizar essa análise, cada experimento foi explorado separadamente (avaliações de  $1,2 \times 10^5$ ,  $6 \times 10^5$  e  $3 \times 10^6$ ).

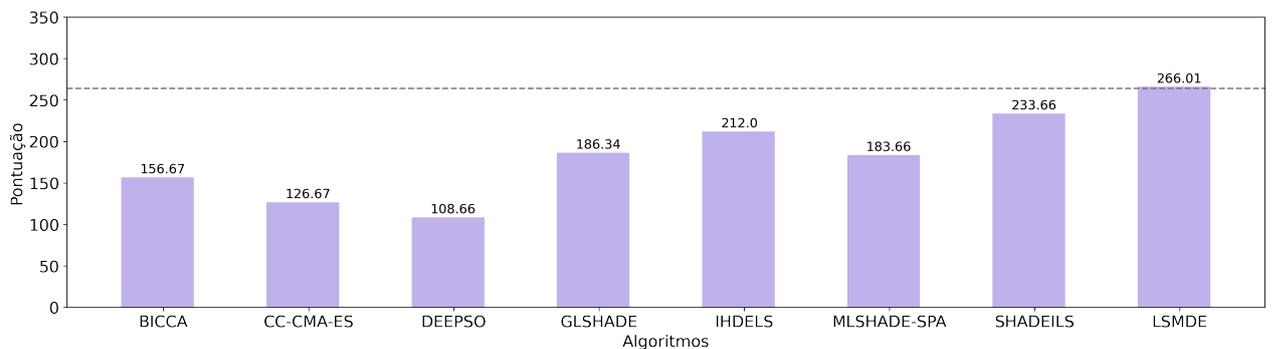


Figura 22 – Pontuação média geral para 8 competidores com base no critério Formula 1 considerando os três experimentos. LSMDE tem o melhor desempenho geral.

Fonte: (FONSECA *et al.*, 2022)

Tabela 6 apresenta os valores médios de aptidão para cada função durante a execução dos experimentos com limite de  $1,2 \times 10^5$ ,  $6 \times 10^5$  e  $3 \times 10^6$  avaliações da função objetivo  $f$ . Com base na Tabela 6, é possível observar que o método proposto LSMDE apresenta o melhor desempenho médio para um experimento com  $1,2 \times 10^5$  e  $6 \times 10^5$  avaliações da função objetivo  $f$ .

Tabela 6 – Valores médios de aptidão para os algoritmos nas funções LSGO do CEC 2013 com  $1,2 \times 10^5$ ,  $6 \times 10^5$  e  $3 \times 10^6$  avaliações.

$f$	FE's	Competidores							
		BICCA	CC-CMA-ES	DEEPSO	GL-SHADE	IHDELS	SHADE-ILS	MLSHADE-SPA	LSMDE
$f_1$	$1,2 \times 10^5$	1,03e+10	1,14e+08	2,99e+10	1,78e+05	2,01e+05	<b>5,12e+04*</b>	6,12e+07	2,15e+06
	$6 \times 10^5$	7,87e+03	3,99e+04	4,61e+09	3,82e+01	4,70e+02	<b>3,55e-23*</b>	1,22e-01	3,50e-09
	$3 \times 10^6$	<b>0,00e+00*</b>	5,80e-09	1,44e+08	3,74e-23	4,34e-28	2,56e-28	1,94e-22	6,88e-23
$f_2$	$1,2 \times 10^5$	2,26e+04	1,40e+03	3,45e+04	<b>6,84e+02*</b>	4,99e+03	2,54e+03	1,74e+03	2,83e+03
	$6 \times 10^5$	<b>1,01e+01*</b>	1,33e+03	2,61e+04	2,35e+01	2,78e+03	1,79e+03	9,13e+01	1,75e+03
	$3 \times 10^6$	<b>8,46e-07*</b>	1,33e+03	1,49e+04	7,76e+00	1,32e+03	1,04e+03	7,89e+01	9,56e+02
$f_3$	$1,2 \times 10^5$	1,75e+01	<b>3,78e-02*</b>	2,12e+01	2,00e+01	2,01e+01	2,01e+01	3,54e+00	2,04e+01
	$6 \times 10^5$	1,92e+00	<b>0,00e+00*</b>	2,09e+01	2,00e+01	2,01e+01	2,01e+01	6,74e-05	2,01e+01
	$3 \times 10^6$	7,27e-01	<b>0,00e+00*</b>	2,04e+01	2,00e+01	2,01e+01	2,01e+01	0,00e+00	2,01e+01
$f_4$	$1,2 \times 10^5$	8,16e+10	6,09e+11	4,27e+11	5,63e+10	<b>2,15e+10*</b>	3,58e+10	6,92e+11	2,38e+10
	$6 \times 10^5$	8,23e+09	3,53e+10	4,95e+10	3,44e+09	2,36e+09	3,74e+09	5,63e+09	<b>2,17e+09*</b>
	$3 \times 10^6$	8,85e+08	2,19e+09	4,77e+09	<b>3,01e+07*</b>	3,04e+08	3,01e+08	6,90e+08	1,96e+08
$f_5$	$1,2 \times 10^5$	6,31e+06	7,28e+14	1,56e+07	4,35e+06	1,24e+07	2,35e+06	1,09e+07	<b>2,26e+06*</b>
	$6 \times 10^5$	3,29e+06	7,28e+14	1,46e+07	2,67e+06	1,02e+07	2,10e+06	2,58e+06	<b>1,75e+06*</b>
	$3 \times 10^6$	2,58e+06	7,28e+14	1,45e+07	2,23e+06	9,59e+06	1,33e+06	1,80e+06	<b>1,15e+06*</b>
$f_6$	$1,2 \times 10^5$	<b>4,82e+05*</b>	7,54e+05	1,05e+06	1,05e+06	1,05e+06	1,05e+06	8,38e+05	1,01e+06
	$6 \times 10^5$	1,87e+05	6,30e+05	1,04e+06	1,05e+06	1,03e+06	1,04e+06	<b>1,60e+03*</b>	1,00e+06
	$3 \times 10^6$	1,46e+05	5,87e+05	1,02e+06	1,03e+06	1,03e+06	1,03e+06	<b>1,40e+03*</b>	9,98e+05
$f_7$	$1,2 \times 10^5$	2,27e+09	5,81e+09	3,50e+09	1,50e+09	3,09e+08	3,68e+08	6,27e+09	<b>1,52e+08*</b>
	$6 \times 10^5$	5,34e+07	1,42e+09	2,40e+08	2,80e+07	1,10e+07	<b>1,54e+06*</b>	1,89e+08	1,05e+07
	$3 \times 10^6$	1,82e+05	7,44e+06	1,54e+07	<b>2,37e+00*</b>	3,46e+04	2,24e+02	5,31e+04	3,91e+04
$f_8$	$1,2 \times 10^5$	1,87e+15	2,93e+16	8,33e+14	4,05e+14	2,09e+14	2,36e+14	2,46e+16	<b>7,11e+11*</b>
	$6 \times 10^5$	3,96e+13	2,30e+15	1,68e+14	1,10e+13	2,22e+13	1,43e+13	4,26e+13	<b>1,36e+11*</b>
	$3 \times 10^6$	3,78e+12	3,88e+14	5,42e+12	1,11e+11	1,36e+12	5,99e+11	9,77e+12	<b>2,68e+09*</b>
$f_9$	$1,2 \times 10^5$	5,73e+08	7,36e+08	1,03e+09	2,55e+09	7,25e+08	2,87e+08	8,23e+08	<b>2,62e+08*</b>
	$6 \times 10^5$	3,19e+08	4,48e+08	9,29e+09	2,40e+09	6,96e+08	2,49e+08	2,15e+08	<b>2,13e+08*</b>
	$3 \times 10^6$	2,18e+08	3,71e+08	9,17e+08	2,36e+09	6,74e+08	1,58e+08	1,61e+08	<b>1,48e+08*</b>
$f_{10}$	$1,2 \times 10^5$	<b>4,00e+06*</b>	2,68e+07	9,41e+07	9,38e+07	9,43e+07	9,39e+07	3,11e+07	9,21e+07
	$6 \times 10^5$	1,89e+06	4,49e+06	9,22e+07	9,27e+07	9,31e+07	9,32e+07	<b>1,11e+03*</b>	9,08e+07
	$3 \times 10^6$	1,24e+06	7,55e+05	9,07e+07	9,17e+07	9,16e+07	9,26e+07	<b>6,56e+02*</b>	9,06e+07
$f_{11}$	$1,2 \times 10^5$	8,48e+10	6,04e+11	7,52e+11	9,44e+11	9,55e+09	5,58e+09	8,67e+11	<b>4,30e+09*</b>
	$6 \times 10^5$	8,59e+08	5,15e+10	8,03e+09	9,27e+11	4,51e+08	<b>1,30e+08*</b>	1,68e+09	6,96e+08
	$3 \times 10^6$	2,85e+07	1,58e+08	5,60e+08	9,27e+11	1,07e+07	<b>5,39e+05*</b>	4,04e+07	1,15e+07
$f_{12}$	$1,2 \times 10^5$	1,20e+11	6,92e+03	7,80e+11	2,42e+04	<b>2,07e+03*</b>	2,64e+03	1,19e+07	4,43e+04
	$6 \times 10^5$	1,67e+04	6,17e+03	1,55e+11	9,60e+02	1,44e+03	1,77e+03	1,02e+03	<b>6,68e+02*</b>
	$3 \times 10^6$	1,40e+03	1,27e+03	1,54e+10	<b>3,19e-01*</b>	3,77e+02	6,49e+01	1,04e+02	3,25e+02
$f_{13}$	$1,2 \times 10^5$	2,11e+10	5,44e+10	1,82e+11	2,69e+10	1,06e+10	1,36e+10	4,29e+10	<b>8,32e+09*</b>
	$6 \times 10^5$	8,24e+08	1,87e+10	1,29e+10	2,73e+09	7,29e+08	<b>5,60e+08*</b>	4,94e+09	1,10e+09
	$3 \times 10^6$	09e+07	6,69e+08	8,75e+08	<b>3,98e+04*</b>	3,80e+06	1,07e+06	7,21e+07	7,12e+06
$f_{14}$	$1,2 \times 10^5$	2,36e+11	8,18e+11	2,18e+12	3,70e+11	8,96e+10	1,84e+11	1,06e+12	<b>5,99e+10*</b>
	$6 \times 10^5$	4,68e+08	1,81e+11	1,63e+11	1,42e+10	<b>1,67e+08*</b>	4,91e+08	2,56e+10	1,11e+09
	$3 \times 10^6$	4,27e+07	7,10e+07	4,33e+08	<b>4,79e+06*</b>	1,58e+07	7,63e+06	1,52e+07	2,54e+07
$f_{15}$	$1,2 \times 10^5$	9,09e+08	1,20e+08	6,60e+12	1,12e+08	5,34e+07	8,86e+07	5,66e+08	<b>4,23e+07*</b>
	$6 \times 10^5$	2,21e+07	4,24e+07	3,00e+07	3,24e+07	<b>1,49e+07</b>	1,69e+07	4,88e+07	2,48e+07
	$3 \times 10^6$	3,16e+06	3,03e+07	7,04e+06	1,29e+06	2,81e+06	<b>8,68e+05*</b>	2,76e+07	6,16e+06
Melhor	$1,2 \times 10^5$	2	1	0	1	2	1	0	<b>8</b>
	$6 \times 10^5$	1	1	0	0	2	4	2	<b>5</b>
	$3 \times 10^6$	1	1	0	<b>5</b>	0	2	3	3

\* Melhor resultado encontrado para a função.

A Figura 23 mostra o desempenho de cada algoritmo baseado no critério Formula 1 para cada categoria de função para  $1,2 \times 10^5$  avaliações. O LSMDE demonstrou seu melhor desempenho em funções com componentes separáveis, funções sem componentes separáveis e funções não separáveis. O LSMDE também apresentou uma pontuação média superior a todos os algoritmos concorrentes, incluindo um dos melhores algoritmos atuais, SHADE-ILS.

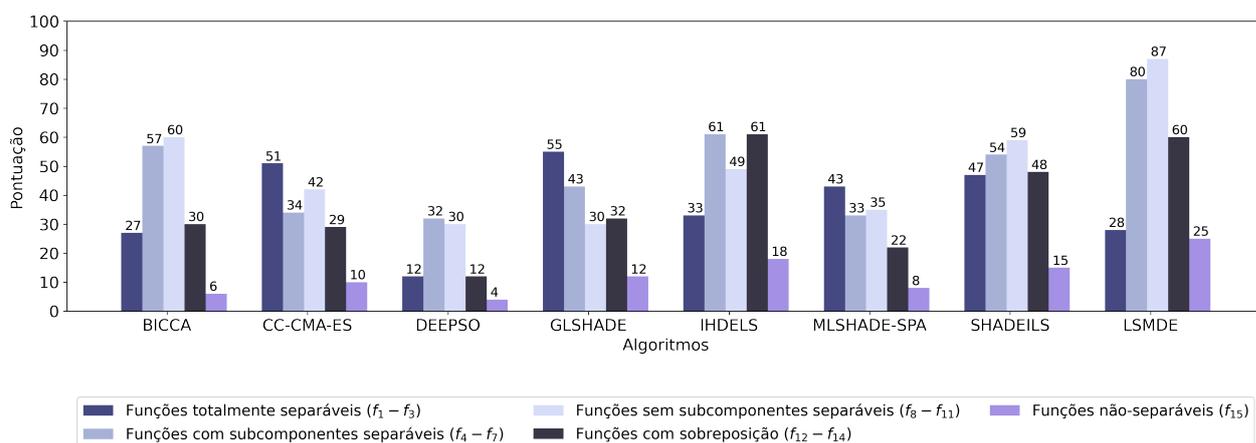


Figura 23 – Pontuação de cada algoritmo concorrente com base no critério Formula 1 para  $1,2 \times 10^5$  avaliações

Fonte: (FONSECA *et al.*, 2022)

Com mais recurso computacional através do aumento do numero de avaliações da função objetivo para  $6 \times 10^5$ , o LSMDE também apresentou melhor desempenho médio do que os outros algoritmos. A Figura 24 reafirma a superioridade do LSMDE em funções com subcomponentes separáveis e funções sem subcomponentes separáveis. Também pode-se observar um desempenho promissor em funções com sobreposição para  $6 \times 10^5$  avaliações.

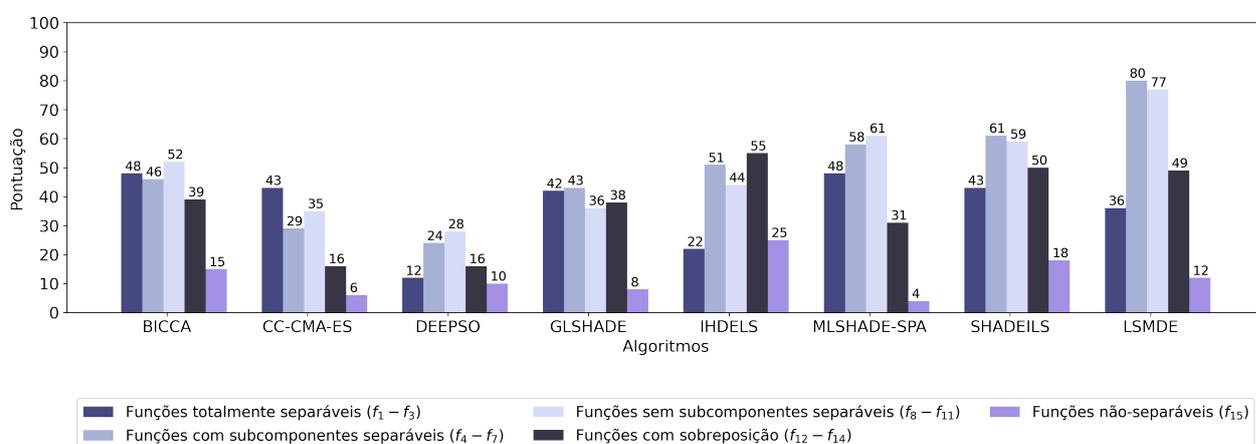


Figura 24 – Pontuação de cada algoritmo concorrente com base no critério Formula 1 para  $6 \times 10^5$  avaliações

Fonte: (FONSECA *et al.*, 2022)

O algoritmo GL-SHADE obteve pontuação mais alta do que outros algoritmos concorrentes para o experimento com um limite máximo de avaliações da função objetivo disponível ( $3 \times 10^6$  avaliações da função objetivo). SHADE-ILS e LSMDE apresentaram a segunda e terceira melhor posição, respectivamente. A Figura 25 mostra que o LSMDE ainda apresenta melhor desempenho em funções com subcomponentes separáveis e funções sem subcomponentes separáveis do que GL-SHADE e SHADE-ILS para  $3 \times 10^6$  avaliações.

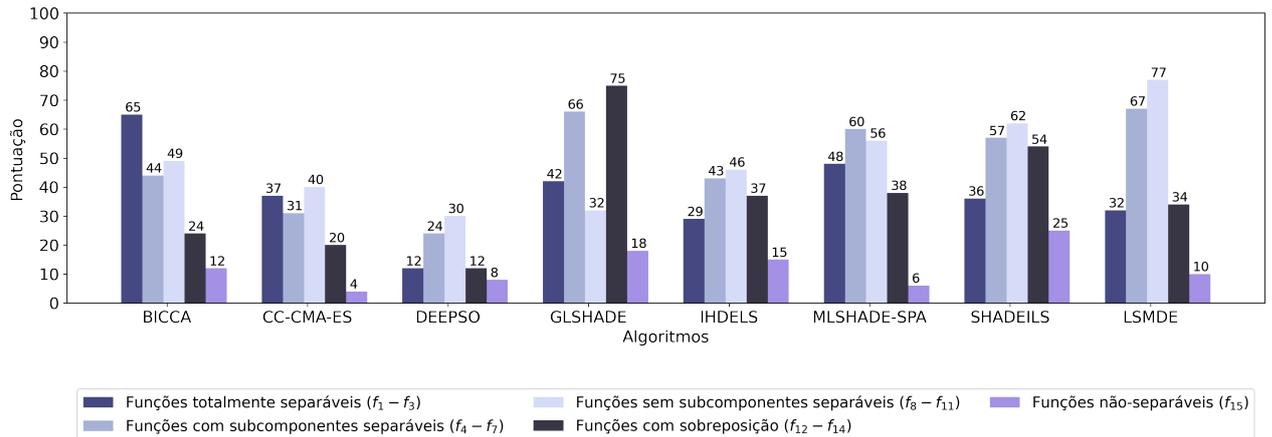


Figura 25 – Pontuação de cada algoritmo concorrente com base no critério Formula 1 para  $3 \times 10^6$  avaliações

Fonte: (FONSECA *et al.*, 2022)

Como dito no começo desta subsecção, a Figura 22 mostra que o método proposto tem a pontuação geral média mais alta. Contudo, de acordo com a Figura 26, percebe-se que o LSMDE tem o melhor desempenho em experimentos com  $1, 2 \times 10^5$  e  $6 \times 10^5$  avaliações da função objetivo, alcançando a terceira melhor posição no experimento com um máximo de  $3 \times 10^6$  avaliações. A Figura 26 apresenta a pontuação geral para cada experimento com a soma das pontuações de cada categoria de função.

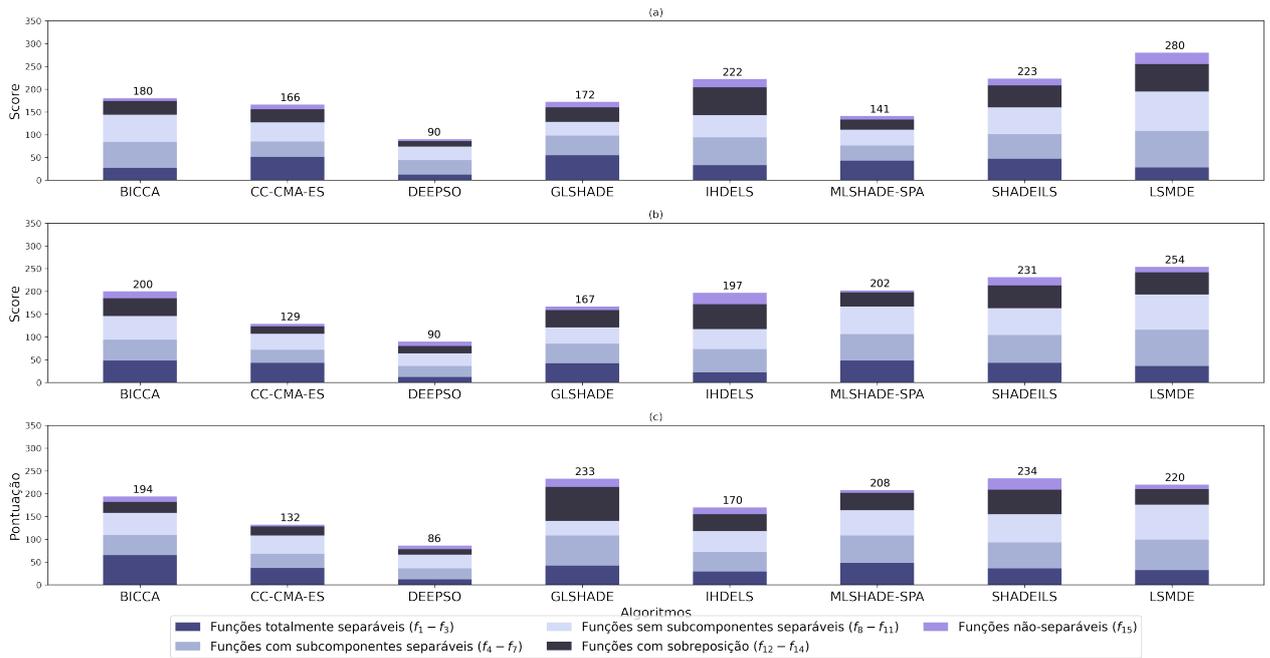


Figura 26 – Pontuação geral baseada no critério Formula 1 para (a)  $1,2 \times 10^5$  avaliações, (b)  $6 \times 10^5$  avaliações e (c)  $3 \times 10^6$  avaliações.

Fonte: (FONSECA *et al.*, 2022)

## 6.2 ANÁLISE DE PERFORMANCE DE ALGORITMOS SIMILARES AO MÉTODO PROPOSTO

Uma segunda rodada de experimentos foi realizada apenas com os algoritmos que possuem características semelhantes ao método proposto nesta tese e com o máximo de recurso computacional disponível através de  $3 \times 10^6$  avaliações da função objetivo. Para fins de comparação, também foram incluídos os três melhores competidores da rodada anterior (GL-SHADE, SHADE-ILS e a proposta de tese, LSMDE). Observa-se pela Tabela 7 que o LSMDE possui resultados superiores aos algoritmos que também utilizam abordagens baseadas em redução de dimensionalidade ou métodos baseados em oposição para a melhor exploração do espaço de alta dimensionalidade. o GL-SHADE, como previsto com base em testes anteriores, continuou a se destacar com o número máximo de avaliações permitidas (como a abordada nesta rodada de experimentos).

Tabela 7 – Valores médios de aptidão para algoritmos com características semelhantes à proposta de tese nas funções LSGO do *CEC'2013* com  $3 \times 10^6$  avaliações.

$f$	FE's	Competidores						
		LS-EDA	MSES	EMT-SAS	GL-SHADE	OJS	SHADE-ILS	LSMDE
$f_1$	$3 \times 10^6$	1,08e+07	1,34e+07	4,05e+06	3,74e-23	1,58e+12	<b>2,56e-28*</b>	6,88e-23
$f_2$	$3 \times 10^6$	1,94e+03	1,80e+04	2,09e+04	<b>7,76e+00*</b>	3,81e+04	1,04e+03	9,56e+02
$f_3$	$3 \times 10^6$	2,16e+01	<b>2,00e+01*</b>	2,15e+01	2,00e+01	2,16e+01	2,01e+01	2,01e+01
$f_4$	$3 \times 10^6$	6,48e+11	1,29e+09	7,19e+08	<b>3,01e+07*</b>	4,45E+13	3,01e+08	1,96e+08
$f_5$	$3 \times 10^6$	8,71e+06	4,52e+06	3,78e+06	2,23e+06	2,85e+07	1,33e+06	<b>1,15e+06*</b>
$f_6$	$3 \times 10^6$	1,06e+06	1,00e+06	1,06e+06	1,03e+06	1,06e+06	1,03e+06	<b>9,98e+05*</b>
$f_7$	$3 \times 10^6$	2,61e+08	1,85e+06	3,78e+06	<b>2,37e+00*</b>	1,10E+13	2,24e+02	3,91e+04
$f_8$	$3 \times 10^6$	2,03e+17	7,14e+12	1,59e+10	1,11e+11	2,04e+18	5,99e+11	<b>2,68e+09*</b>
$f_9$	$3 \times 10^6$	6,58e+08	4,48e+08	4,14e+08	2,36e+09	1,98e+09	1,58e+08	<b>1,48e+08*</b>
$f_{10}$	$3 \times 10^6$	9,40e+07	9,10e+07	9,34e+07	9,17e+07	<b>3,79e+04*</b>	9,26e+07	9,06e+07
$f_{11}$	$3 \times 10^6$	1,69e+11	1,51e+07	5,30e+07	9,27e+11	5,95e+14	<b>5,39e+05*</b>	1,15e+07
$f_{12}$	$3 \times 10^6$	1,15e+03	1,68e+03	3,64e+03	<b>3,19e-01*</b>	1,67e+12	6,49e+01	3,25e+02
$f_{13}$	$3 \times 10^6$	1,46e+09	1,51e+07	2,34e+07	<b>3,98e+04*</b>	2,07e+08	1,07e+06	7,12e+06
$f_{14}$	$3 \times 10^6$	3,33e+09	4,83e+07	4,08e+07	<b>4,79e+06*</b>	6,65e+15	7,63e+06	2,54e+07
$f_{15}$	$3 \times 10^6$	8,06e+06	1,64e+06	4,02e+06	1,29e+06	4,59e+09	<b>8,68e+05*</b>	6,16e+06
Melhor	$3 \times 10^6$	0	1	0	6	1	3	4

\* Melhor resultado encontrado para a função.

### 6.3 ANÁLISE ESTATÍSTICA DO LSMDE COMPARADA COM O ESTADO DA ARTE

Para garantir a confiança estatística dos resultados, foram selecionados os métodos de melhor desempenho nos experimentos anteriores (SHADE-ILS, GL-SHADE, LSMDE) para realizar uma análise mais detalhada. De acordo com [Soria-Alcaraz Jorge et al. \(2013\)](#), o teste não paramétrico de Kruskal-Wallis é recomendado para comparar o desempenho entre três algoritmos bioinspirados. No entanto, o teste estatístico indica apenas se há ou não diferença estatística entre os três competidores comparados. Esta tese utiliza dois testes *post hoc*, Conover e Dunn, com correção de Holm para determinar qual algoritmo tem o melhor desempenho ([OSTERTAGOVA; OSTERTAG; KOVÁČ, 2014](#)).

A Tabela 8 e 9 apresentam os  $p$ -valores do LSMDE versus os melhores competidores anteriores para os testes *post hoc* de Dunn e Conover, respectivamente.  $p$ -valores menores que 0,05 indicam diferença estatística entre o controle e o algoritmo comparado para a função atual. Neste caso, o LSMDE é o controle. Os  $p$ -valores em negrito representam funções em que o LSMDE teve desempenho estatisticamente superior ao seu adversário (SHADE-ILS ou GL-SHADE). Os outros  $p$ -valores representam funções em que o LSMDE é estatisticamente igual ou inferior ao seu adversário.

As Tabelas 8 e 9 resumem os resultados da comparação do método proposto

Tabela 8 – Resultados do teste de Dunn,  $p$ -valores do LSMDE versus SHADE-ILS e GL-SHADE nas funções de referência CEC'2013 para 1000 dimensões. Os símbolos "+", "-" e "=" denotam que o LSMDE é melhor que, pior que ou similar ao algoritmo comparado, respectivamente.

Funções	Experimento 1: $1,2 \times 10^5$		Experimento 2: $6 \times 10^5$		Experimento 3: $3 \times 10^6$	
	SHADE-ILS	GL-SHADE	SHADE-ILS	GL-SHADE	SHADE-ILS	GL-SHADE
$f_1$	5,63e-30 (-)	1,38e-08 (-)	1,74e-08 (-)	<b>1,74e-08 (+)</b>	6,76e-11 (-)	<b>5,49e-05 (+)</b>
$f_2$	1,73e-01 (=)	3,72e-15 (-)	<b>2,48e-05 (+)</b>	1,38e-10 (-)	<b>4,54e-06 (+)</b>	4,62e-10 (-)
$f_3$	2,81e-02 (-)	6,77e-22 (-)	1,17e-01 (=)	1,39e-20 (-)	3,00e-01 (=)	3,68e-20 (-)
$f_4$	7,65e-01 (=)	<b>1,04e-11 (+)</b>	7,04e-02 (=)	<b>2,49e-04 (+)</b>	2,91e-02 (-)	2,13e-21 (-)
$f_5$	6,76e-02 (=)	<b>2,37e-14 (+)</b>	<b>1,06e-02 (+)</b>	<b>2,78e-16 (+)</b>	2,68e-01 (=)	<b>8,01e-19 (+)</b>
$f_6$	<b>9,36e-14 (+)</b>	<b>6,08e-18 (+)</b>	<b>6,96e-10 (+)</b>	<b>1,32e-27 (+)</b>	<b>8,05e-16 (+)</b>	<b>2,15e-19 (+)</b>
$f_7$	<b>7,58e-08 (+)</b>	<b>6,88e-29 (+)</b>	7,61e-09 (-)	<b>3,07e-07 (+)</b>	4,49e-13 (-)	1,14e-22 (-)
$f_8$	<b>5,75e-09 (+)</b>	<b>2,98e-29 (+)</b>	<b>2,97e-13 (+)</b>	<b>2,12e-22 (+)</b>	<b>2,41e-19 (+)</b>	<b>4,40e-15 (+)</b>
$f_9$	7,98e-01 (=)	<b>8,77e-18 (+)</b>	<b>1,65e-03 (+)</b>	<b>2,29e-16 (+)</b>	2,30e-01 (=)	<b>1,14e-19 (+)</b>
$f_{10}$	<b>3,66e-06 (+)</b>	<b>1,02e-12 (+)</b>	<b>3,57e-12 (+)</b>	<b>6,45e-24 (+)</b>	<b>7,07e-19 (-)</b>	<b>2,73e-16 (+)</b>
$f_{11}$	4,81e-03 (-)	<b>1,43e-12 (+)</b>	2,17e-20 (-)	1,30e-01 (=)	3,00e-09 (-)	9,90e-29 (-)
$f_{12}$	1,43e-15 (-)	2,59e-01 (=)	<b>6,46e-18 (+)</b>	<b>4,64e-06 (+)</b>	6,51e-06 (-)	2,27e-22 (-)
$f_{13}$	4,20e-01 (=)	<b>2,09e-15 (+)</b>	7,72e-09 (-)	<b>1,10e-07 (+)</b>	5,50e-14 (-)	1,43e-21 (-)
$f_{14}$	8,48e-01 (=)	<b>7,48e-17 (+)</b>	4,10e-09 (-)	<b>1,60e-07 (+)</b>	2,32e-09 (-)	1,58e-28 (-)
$f_{15}$	<b>7,44e-10 (+)</b>	<b>2,32e-22 (+)</b>	8,89e-13 (-)	1,31e-01 (=)	3,42e-20 (-)	5,92e-15 (-)
$m/i/p^*$	5/6/4	11/1/3	7/2/6	11/2/2	4/3/8	6/0/9

\* LSMDE é significativamente melhor em  $m$  funções, igual em  $i$  funções e pior em  $p$  funções.

LSMDE com as melhores metaheurísticas para LSGO, SHADE-ILS e GL-SHADE. Esses resultados podem ser analisados considerando cada experimento separadamente: experimento 1 com  $1,2 \times 10^5$  avaliações, experimento 2 com  $6 \times 10^5$  avaliações e experimento 3 com  $3 \times 10^6$  avaliações.

Com base nos experimentos com  $1,2 \times 10^5$  e  $6 \times 10^5$  avaliações das Tabelas 8 e 9, observa-se que o método proposto é estatisticamente superior ao GL-SHADE em 11/12 das 15 funções do CEC'2013. O GL-SHADE superou o LSMDE em 9 funções para  $3 \times 10^6$  avaliações. No entanto, o LSMDE teve um desempenho melhor em funções parcialmente separáveis, superando o GL-SHADE em 5 das 8 funções disponíveis nesta categoria (Figura 27). Ao comparar o LSMDE e o SHADE-ILS, observa-se uma melhora no desempenho do método proposto nos experimentos com  $1,2 \times 10^5$  e  $6 \times 10^5$  avaliações. Essa melhora é mais evidente quando analisa-se a categoria de funções parcialmente separáveis ( $f_4 - f_{11}$ ), como observado na Figura 27. Para experimento  $3 \times 10^6$  avaliações, observa-se que o LSMDE é estatisticamente igual ou superior em 7 funções e estatisticamente inferior ao SHADE-ILS em 8 funções. A Figura 27 mostra que o LSMDE obteve melhor desempenho em 4 funções para problemas parcialmente separáveis no experimento com  $3 \times 10^6$  avaliações, de acordo com o teste de Conover.

Tabela 9 – Resultados do teste de Conover,  $p$ -valores de LSMDE versus SHADE-ILS e GL-SHADE nas funções de *benchmark* CEC 2013 para 1000 dimensões. Os símbolos "+", "-" e "=" indicam que o LSMDE é melhor, pior ou similar ao algoritmo comparado, respectivamente.

Funções	Experimento 1: $1,2 \times 10^5$		Experimento 2: $6 \times 10^5$		Experimento 3: $3 \times 10^6$	
	SHADE-ILS	GL-SHADE	SHADE-ILS	GL-SHADE	SHADE-ILS	GL-SHADE
$f_1$	7,15e-70 (-)	9,07e-36 (-)	1,34e-36 (-)	<b>1,34e-36 (+)</b>	1,77e-28 (-)	<b>1,89e-14 (+)</b>
$f_2$	<b>1,81e-02 (+)</b>	2,22e-28 (-)	<b>7,71e-16 (+)</b>	1,49e-28 (-)	<b>2,67e-19 (+)</b>	1,75e-29 (-)
$f_3$	1,10e-04 (-)	1,40e-37 (-)	6,43e-03 (-)	4,06e-35 (-)	6,23e-02 (=)	9,62e-36 (-)
$f_4$	6,90e-01 (=)	<b>6,41e-16 (+)</b>	5,73e-02 (=)	<b>1,82e-04 (+)</b>	1,81e-04 (-)	7,45e-36 (-)
$f_5$	1,40e-03 (-)	<b>7,10e-28 (+)</b>	<b>5,31e-04 (+)</b>	<b>7,89e-22 (+)</b>	6,74e-02 (=)	<b>7,91e-31 (+)</b>
$f_6$	<b>5,36e-23 (+)</b>	<b>6,88e-28 (+)</b>	<b>6,84e-30 (+)</b>	<b>3,40e-55 (+)</b>	<b>6,67e-29 (+)</b>	<b>3,68e-33 (+)</b>
$f_7$	<b>2,69e-28 (+)</b>	<b>5,86e-62 (+)</b>	1,62e-27 (-)	<b>4,28e-23 (+)</b>	2,88e-27 (-)	3,63e-39 (-)
$f_8$	<b>2,64e-33 (+)</b>	<b>1,86e-64 (+)</b>	<b>2,92e-27 (+)</b>	<b>1,51e-38 (+)</b>	<b>2,95e-32 (+)</b>	<b>3,39e-27 (+)</b>
$f_9$	6,64e-01 (=)	<b>1,43e-30 (+)</b>	<b>2,90e-05 (+)</b>	<b>1,37e-21 (+)</b>	<b>3,97e-02 (+)</b>	<b>2,72e-33 (+)</b>
$f_{10}$	<b>3,64e-08 (+)</b>	<b>1,92e-15 (+)</b>	<b>3,00e-27 (+)</b>	<b>3,31e-42 (+)</b>	<b>2,12e-32 (+)</b>	<b>2,54e-29 (+)</b>
$f_{11}$	5,75e-07 (-)	<b>1,01e-26 (+)</b>	1,23e-34 (-)	8,86e-03 (-)	5,87e-32 (-)	3,87e-61 (-)
$f_{12}$	1,07e-28 (-)	5,12e-02 (=)	<b>2,43e-24 (+)</b>	<b>6,90e-10 (+)</b>	3,48e-12 (-)	6,30e-35 (-)
$f_{13}$	1,85e-01 (=)	<b>1,30e-26 (+)</b>	1,72e-30 (-)	<b>5,46e-27 (+)</b>	1,07e-27 (-)	6,25e-37 (-)
$f_{14}$	7,56e-01 (=)	<b>8,69e-28 (+)</b>	6,95e-32 (-)	<b>3,52e-27 (+)</b>	1,65e-31 (-)	5,89e-60 (-)
$f_{15}$	<b>5,50e-20 (+)</b>	<b>8,40e-36 (+)</b>	3,92e-21 (-)	<b>2,09e-02 (+)</b>	3,01e-34 (-)	5,62e-28 (-)
$m/i/p^*$	6/4/5	11/1/3	7/1/7	12/0/3	5/2/8	6/0/9

\* LSMDE é significativamente melhor em  $m$  funções, igual em  $i$  funções e pior em  $p$  funções.

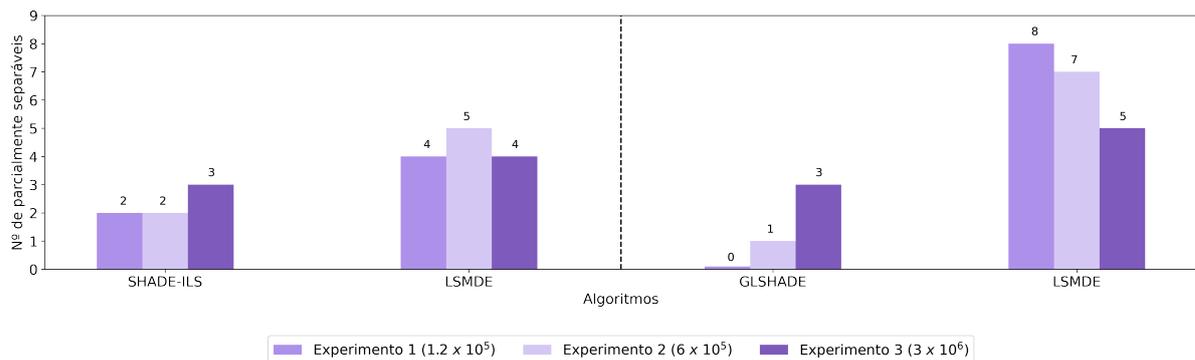


Figura 27 – Número das melhores funções parcialmente separáveis. LSMDE vs SHADE-ILS e LSMDE vs GL-SHADE de acordo com o teste *post-hoc* de Conover para  $1,2 \times 10^5$ ,  $6 \times 10^5$  e  $3 \times 10^6$  avaliações.

Fonte: (FONSECA *et al.*, 2022)

As análises usando o critério Formula 1 sugerido pela *IEEE Task Force on Large-Scale Global Optimization* demonstram que o LSMDE tem um desempenho melhor do que outros algoritmos de última geração. No entanto, o teste estatístico Kruskal-Wallis permite avaliar qual função cada algoritmo concorrente tem o desempenho similar, melhor ou pior. Como cada função representa uma ampla gama de problemas

de otimização do mundo real, o teste estatístico permite escolher a metaheurística mais adequada para cada situação. Portanto, o método proposto tem um desempenho melhor, especialmente em problemas parcialmente separáveis. Também podem-se concluir que o LSMDE é competitivo com os melhores algoritmos usados nesta tese.

#### 6.4 ANÁLISE DE ESCALABILIDADE DO LSMDE COMPARADA COM O ESTADO DA ARTE

Na Figura 28, pode-se observar o desempenho do LSMDE, SHADE-ILS e GL-SHADE sobre a agregação de todas as funções à medida que a dimensionalidade aumenta na suite de teste *bbob-largescale*. Observa-se que o LSMDE tem uma taxa de sucesso entre 40% e 80%, enquanto o SHADE-ILS tem uma taxa de sucesso entre 38% e 60% e o GL-SHADE apresentou uma taxa de sucesso entre 20% e 50%. Este resultado mostra que o LSMDE é mais eficaz na resolução dos problemas propostos quando há um aumento na dimensionalidade.

As Figuras 29a a 29f mostram o gráfico ECDF para algumas funções selecionadas com diferentes características. Pode-se ver o bom desempenho do LSMDE na função Rastrigin separável (Figura 29a), mesmo para altas dimensionalidades. No entanto, o LSMDE converge para soluções piores na versão não separável da função Rastrigin (Figura 29c). Nesse caso, o número de ótimos locais é aproximadamente  $10^d$  ( $d$  é a dimensionalidade do problema) (FINCK *et al.*, 2010). Concluí-se que a função não separável de Rastrigin é muito complexa para o LSMDE resolver sem ajustes adicionais nos hiperparâmetros.

Em relação às funções Rosenbrock (Figura 29d e 29e), observa-se bons resultados, mesmo com o aumento da dimensionalidade e adições de complexidade (como rotações e baixo/moderado condicionamento). Uma exceção ocorre quando a dimensionalidade aumenta de 320 para 640 nas funções Rosenbrock rotacionadas e de setor atrativo (Figura 29e e Figura 29b), onde percebe-se uma degradação significativa no desempenho. Os resultados indicam que há um limite na capacidade do LSMDE de manter a qualidade das soluções candidatas no espaço reduzido influenciado pelas características intrínsecas do espaço de busca. Na Figura 29f, destaca-se os resultados obtidos na função *Different-Powers*, tanto em dimensões baixas quanto altas. Uma característica interessante observada nesta função é a necessidade de mais avaliações de função para resolver o problema em dimensões mais altas.

As ECDF's para os três algoritmos selecionados, agregados em todas as 24 funções *bbob-largescale*, podem ser encontradas na Figura 30 para dimensões 20-D, 40-D, 80-D, 160-D, 320-D e 640-D. Com base na Figura 30, concluí-se que:

- Em todos os problemas *bbob-largescale*, os algoritmos têm um desempenho pior com o aumento da dimensionalidade, o que já era esperado;

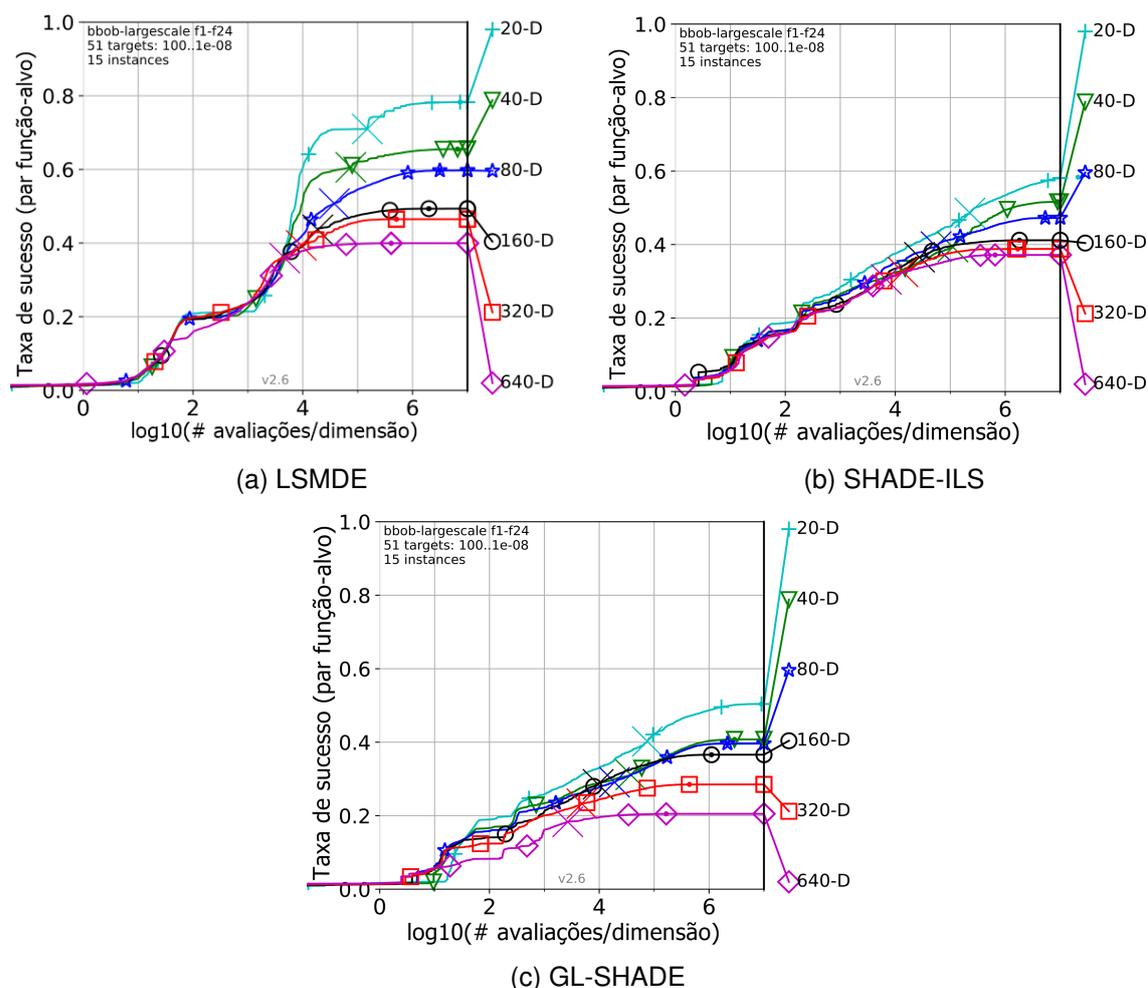


Figura 28 – Gráfico ECDF mostrando o desempenho do (a) LSMDE, (b) SHADE-ILS e (c) GL-SHADE, para diferentes proporções de número de avaliações/dimensões em todos os 24 problemas da suite de teste *bbob-largescale*.

Fonte: (FONSECA *et al.*, 2023)

- GL-SHADE é o mais afetado pelo aumento na dimensionalidade, de tal maneira que apenas cerca de 20% dos valores-alvo foram alcançados em comparação com cerca dos 40% alcançados pelo LSMDE para 640-D;
- LSMDE supera todos os algoritmos concorrentes com o aumento da dimensionalidade para diferentes categorias de funções; no entanto, à medida que a dimensionalidade aumenta, o desempenho tanto do LSMDE quanto do SHADE-ILS se torna semelhante, com uma diferença de desempenho de aproximadamente 5% para 640-D.

Um aspecto importante a destacar é o baixo desempenho do LSMDE e de todos os algoritmos concorrentes nas funções de estrutura global fraca para 160-D, 320-D e 640-D (máximo de 20% de sucesso), como mostrado na Figura 31. Uma função tem uma estrutura global fraca se as aptidões (os valores da função objetivo) dos ótimos locais estão apenas fracamente relacionados às suas localizações no espaço de busca.

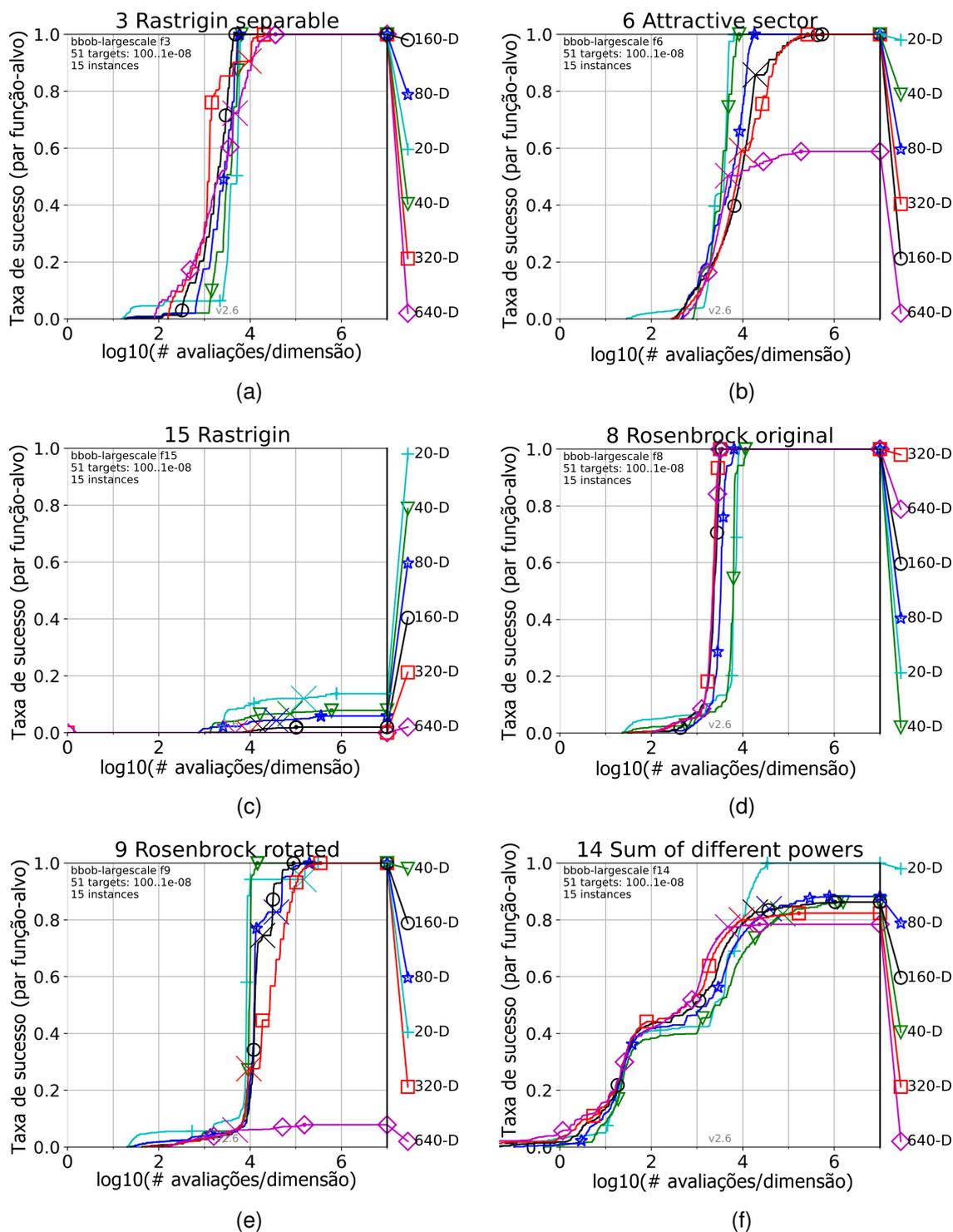


Figura 29 – Gráficos ECDF do LSMDE para as funções  $f_3, f_6, f_8, f_9, f_{14}$  and  $f_{15}$  no conjunto bbob-largescale, agregados em todas as instâncias de cada função.

Fonte: (FONSECA *et al.*, 2023)

Por exemplo, quando os ótimos vizinhos geralmente não têm aptidões semelhantes (BROCKHOFF *et al.*, 2022). Esse comportamento torna a busca por uma solução ótima mais difícil, especialmente em alta dimensionalidade. Como o LSMDE usa um mapeamento contínuo do espaço de busca de alta dimensionalidade para um espaço

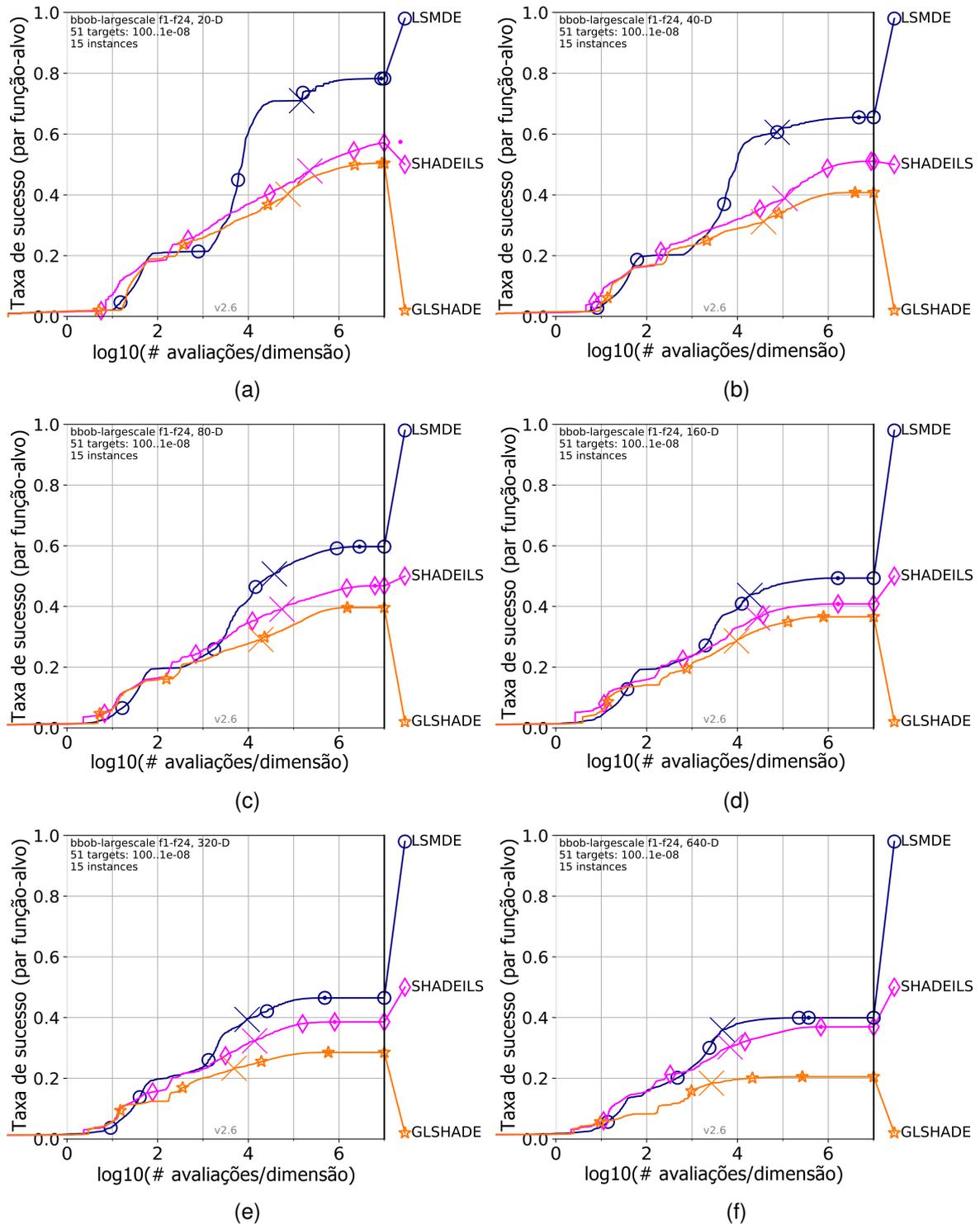


Figura 30 – ECDF para os três algoritmos selecionados, agregados em todas as 24 funções do conjunto *bbob-largescale*.

Fonte: (FONSECA *et al.*, 2023)

de baixa dimensionalidade, as funções de estrutura global fraca não puderam produzir um espaço de busca alternativo que garantisse a posição das soluções candidatas, o que causou uma deterioração na qualidade da exploração do espaço reduzido gerado. Os resultados sob essa classe específica de funções abre a oportunidade de novas

pesquisas e melhorias do método proposto.

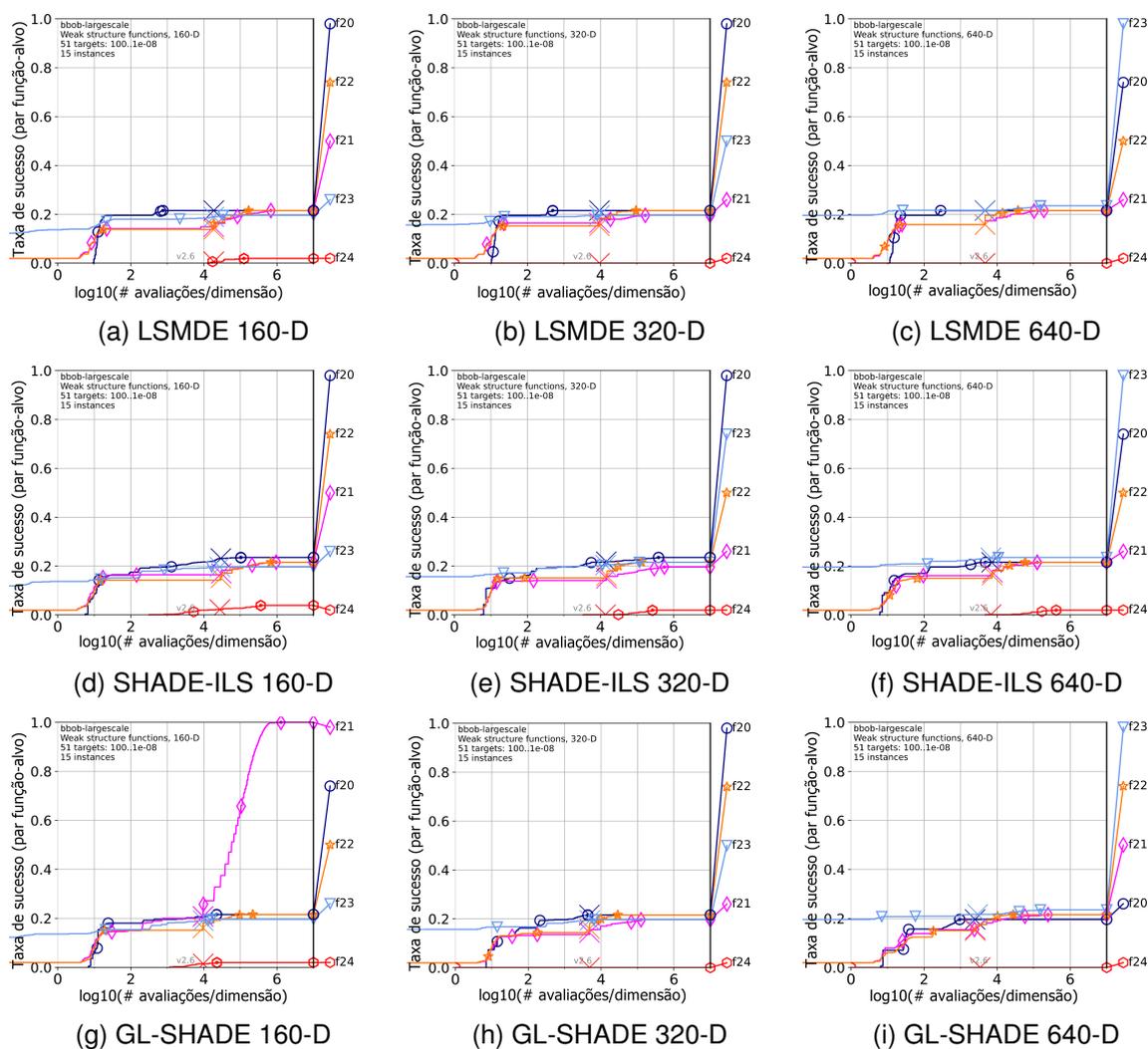


Figura 31 – ECDF para o grupo de estruturas globais fracas funciona com 160-D, 320-D e 640-D.

Fonte: (FONSECA *et al.*, 2023)

## 7 CONSIDERAÇÕES FINAIS

Nesta tese, foi proposto um novo método denominado *Low-dimensional Space Modeling-based Differential Evolution* (LSMDE) para resolver problemas de Otimização Global em Grande Escala. O LSMDE utiliza um algoritmo de Decomposição de Valor Singular para reduzir o espaço de busca  $d$ -dimensional de acordo com os recursos da população atual de soluções candidatas. A busca por soluções promissoras no espaço de baixa dimensão é guiada por um algoritmo SHADE hibridizado com modelos de mistura gaussiana. Este modelo permite o agrupamento de soluções de baixa dimensão, levando em consideração a incerteza inerente a transformação de espaço e mitigando a perda de informações nesse processo. Candidatos para a solução ótima podem ser retornados para o espaço original, e a vizinhança ao redor do ótimo pode ser explorada adequadamente por meio de algoritmos de busca local especializados. O método proposto foi testado a partir de métricas de performance e escalabilidade, utilizando os principais *benchmarks* da literatura, *CEC'2013* e *bbob-largescale*.

O Teorema "*No Free Lunch*" prova matematicamente que nenhum algoritmo ou método pode ser o melhor para todos os problemas (WOLPERT; MACREADY, 1997). No entanto, de acordo com os resultados experimentais, o LSMDE apresenta vantagens relevantes como:

- i. desempenho superior em funções parcialmente separáveis, indicando maior aplicabilidade em problemas do mundo real;
- ii. ótimo desempenho geral em experimentos com recursos computacionais limitados, mostrando-se eficiente mesmo em cenários restritos;
- iii. método de redução de dimensionalidade inovador que pode ser adaptado e aprimorado para diferentes algoritmos populacionais, demonstrando sua versatilidade;
- iv. flexibilidade e aplicabilidade a diversos problemas de alta dimensionalidade, graças à aprendizagem com a população de soluções candidatas sem conhecimento prévio das características do espaço de busca;
- v. melhor escalabilidade do que as principais metaheurísticas da literatura, apresentando uma perda de performance menor com o aumento da dimensionalidade do problema, independente das características do espaço de busca;
- vi. menor número de hiper-parâmetros necessários: A incorporação do modelo de redução de dimensionalidade e do modelo de mistura gaussiana não resultou em um aumento significativo no número de hiper-parâmetros. Essa característica simplifica a configuração do LSMDE e pode contribuir para uma implementação mais direta e menos propensa a erros de ajuste de parâmetros.

No entanto, também foram identificadas algumas limitações, como a falta de ganho substancial quando há disponibilidade de grandes recursos computacionais (representado nessa tese pelo número de avaliações da função objetivo). Além disso, o desempenho do LSMDE em funções não-separáveis e com estrutura global fraca foi limitado, sugerindo a necessidade de ajustes nos hiper-parâmetros do algoritmo para esses problemas.

Os resultados obtidos indicam que o LSMDE é um método evolutivo promissor e prático para resolver problemas de otimização, especialmente quando a dimensionalidade do problema aumenta. Entretanto, pesquisas futuras são necessárias para melhorar seu desempenho em funções desafiadoras com estruturas globais fracas.

Para trabalhos futuros, o LSMDE pode ser aprimorado por meio do ajuste fino dos parâmetros de desempenho, a fim de resolver melhor funções não-separáveis. Além disso, mapeamentos alternativos ou diferentes representações do espaço de busca podem ser explorados para enfrentar o desafio imposto por funções com estruturas globais fracas. A aplicação do LSMDE em diversas áreas, como otimização de parâmetros, otimização multiobjetivo, mineração de dados e seleção de atributos, também é um objetivo a ser perseguido.

## REFERÊNCIAS

BAGATTINI, Francesco; SCHOEN, Fabio; TIGLI, Luca. Clustering methods for large scale geometrical global optimization. **Optimization Methods and Software**, Taylor & Francis, v. 34, n. 5, p. 1099–1122, 2019. Citado nas pp. 20, 21, 31, 40, 42.

BAGATTINI, Francesco; SCHOEN, Fabio; TIGLI, Luca. Clustering methods for the optimization of atomic cluster structure. **The Journal of chemical physics**, AIP Publishing LLC, v. 148, n. 14, p. 144102, 2018. Citado nas pp. 20, 21, 31, 40, 42.

BISHOP, Christopher M; NASRABADI, Nasser M. **Pattern recognition and machine learning**. [S.l.]: Springer, 2006. v. 4. Citado nas pp. 60, 65.

BLEI, David M; JORDAN, Michael I. Variational inference for Dirichlet process mixtures. **Bayesian analysis**, International Society for Bayesian Analysis, v. 1, n. 1, p. 121–143, 2006. Citado na p. 84.

BOLUFÉ-RÖHLER, Antonio; CHEN, Stephen; TAMAYO-VERA, Dania. An analysis of minimum population search on large scale global optimization. *In*: IEEE. 2019 IEEE Congress on Evolutionary Computation (CEC). [S.l.: s.n.], 2019. P. 1228–1235. Citado na p. 20.

BOUKHARI, Nouredine *et al.* A study on self-adaptation in the evolutionary strategy algorithm. *In*: SPRINGER. COMPUTATIONAL Intelligence and Its Applications: 6th IFIP TC 5 International Conference, CIIA 2018, Oran, Algeria, May 8-10, 2018, Proceedings 6. [S.l.: s.n.], 2018. P. 150–160. Citado na p. 45.

BROCKHOFF, Dimo *et al.* Using well-understood single-objective functions in multiobjective black-box optimization test suites. **Evolutionary computation**, MIT Press, v. 30, n. 2, p. 165–193, 2022. Citado na p. 109.

CARAFFINI, Fabio; NERI, Ferrante; PICINALI, Lorenzo. An analysis on separability for memetic computing automatic design. **Information Sciences**, Elsevier, v. 265, p. 1–22, 2014. Citado na p. 19.

CHEN, An *et al.* A global information based adaptive threshold for grouping large scale optimization problems. *In*: PROCEEDINGS of the Genetic and Evolutionary Computation Conference. [S.l.: s.n.], 2018. P. 833–840. Citado nas pp. 31, 34.

- CHEN, Maiyue; TAN, Ying. SF-FWA: A Self-Adaptive Fast Fireworks Algorithm for effective large-scale optimization. **Swarm and Evolutionary Computation**, Elsevier, v. 80, p. 101314, 2023. Citado nas pp. 19, 30.
- CHEN, Qidong; SUN, Jun; PALADE, Vasile. Distributed contribution-based quantum-behaved particle swarm optimization with controlled diversity for large-scale global optimization problems. **IEEE Access**, IEEE, v. 7, p. 150093–150104, 2019. Citado nas pp. 31, 35.
- CHEN, Ying-ping *et al.* A survey of linkage learning techniques in genetic and evolutionary algorithms. **IlligAL report**, Citeseer, v. 2007014, 2007. Citado na p. 19.
- CHOWDHARY, Chiranjil Lal *et al.* Analytical study of hybrid techniques for image encryption and decryption. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 20, n. 18, p. 5162, 2020. Citado na p. 19.
- CONSTANTINE, PG. Active subspaces, SIAM spotlights, vol. 2. **Society for industrial and applied mathematics (SIAM), Philadelphia, PA. Emerging ideas for dimension reduction in parameter studies**, 2015. Citado na p. 66.
- D'ORTENZIO, Alessandro; MANES, Costanzo. Consistency issues in Gaussian Mixture Models reduction algorithms. **arXiv preprint arXiv:2104.12586**, 2021. Citado na p. 62.
- DAS, Swagatam; SUGANTHAN, Ponnuthurai Nagaratnam. Differential evolution: A survey of the state-of-the-art. **IEEE transactions on evolutionary computation**, IEEE, v. 15, n. 1, p. 4–31, 2010. Citado na p. 45.
- DASGUPTA, Sanjoy; GUPTA, Anupam. An elementary proof of a theorem of Johnson and Lindenstrauss. **Random Structures & Algorithms**, Wiley Online Library, v. 22, n. 1, p. 60–65, 2003. Citado na p. 57.
- DE FALCO, Ivanoe; DELLA CIOPPA, Antonio; TRUNFIO, Giuseppe A. Investigating surrogate-assisted cooperative coevolution for large-scale global optimization. **Information Sciences**, Elsevier, v. 482, p. 1–26, 2019. Citado nas pp. 19–21, 30, 32, 45.
- DEL SER, Javier *et al.* Bio-inspired computation: Where we stand and what's next. **Swarm and Evolutionary Computation**, Elsevier, v. 48, p. 220–250, 2019. Citado nas pp. 20, 37.

DEMPSTER, Arthur P; LAIRD, Nan M; RUBIN, Donald B. Maximum likelihood from incomplete data via the EM algorithm. **Journal of the royal statistical society: series B (methodological)**, Wiley Online Library, v. 39, n. 1, p. 1–22, 1977. Citado na p. 63.

DENG, Hanbo *et al.* Ranking-based biased learning swarm optimizer for large-scale optimization. **Information Sciences**, Elsevier, v. 493, p. 120–137, 2019. Citado nas pp. 30, 38.

DENG, Wu *et al.* An adaptive differential evolution algorithm based on belief space and generalized opposition-based learning for resource allocation. **Applied Soft Computing**, Elsevier, v. 127, p. 109419, 2022. Citado na p. 49.

DONG, Wenyong; WANG, Yufeng; ZHOU, Mengchu. A latent space-based estimation of distribution algorithm for large-scale global optimization. **Soft Computing**, Springer, v. 23, n. 13, p. 4593–4615, 2019. Citado nas pp. 31, 40, 42, 94.

ELHARA, Ouassim *et al.* COCO: the large scale black-box optimization benchmarking (BBOB-largescale) test suite. **arXiv preprint arXiv:1903.06396**, 2019. Citado na p. 93.

FENG, Liang *et al.* Multi-space evolutionary search for large-scale optimization. **arXiv preprint arXiv:2102.11693**, 2021. Citado nas pp. 30, 40, 42, 94.

FINCK, Steffen *et al.* **Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions**. [S.l.], 2010. Citado na p. 107.

FISTER, Iztok; ŽUMER, Janez BresViljem *et al.* Memetic artificial bee colony algorithm for large-scale global optimization. *In: IEEE. 2012 IEEE Congress on evolutionary computation*. [S.l.: s.n.], 2012. P. 1–8. Citado nas pp. 31, 37.

FONSECA, Thiago Henrique Lemos *et al.* Low-dimensional Space Modeling-based Differential Evolution for Large Scale Global Optimization Problems. **IEEE Transactions on Evolutionary Computation**, p. 1–1, 2022. DOI: 10.1109/TEVC.2022.3227440. Citado nas pp. 23, 30, 42, 79, 80, 86, 87, 99, 101–103, 106.

FONSECA, Thiago Henrique Lemos *et al.* Low-Dimensional Space Modeling-Based Differential Evolution: A Scalability Perspective on bbob-largescale suite. *In: ROJAS, Ignacio; JOYA, Gonzalo; CATALA, Andreu (Ed.). Advances in Computational Intelligence*. Cham: Springer Nature Switzerland, 2023. P. 16–28. Citado nas pp. 23, 108–111.

GAUVAIN, J-L; LEE, Chin-Hui. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. **IEEE transactions on speech and audio processing**, IEEE, v. 2, n. 2, p. 291–298, 1994. Citado na p. 60.

GE, Hongwei; SUN, Liang *et al.* Cooperative differential evolution with fast variable interdependence learning and cross-cluster mutation. **Applied Soft Computing**, Elsevier, v. 36, p. 300–314, 2015. Citado nas pp. 31, 34.

GE, Hongwei; ZHAO, Mingde *et al.* Bi-space Interactive Cooperative Coevolutionary algorithm for large scale black-box optimization. **Applied Soft Computing**, Elsevier, v. 97, p. 106798, 2020. Citado nas pp. 30, 36, 42, 94.

GLORIEUX, Emile *et al.* Constructive cooperative coevolution for large-scale global optimisation. **Journal of Heuristics**, Springer, v. 23, n. 6, p. 449–469, 2017. Citado nas pp. 31, 33.

GOLUB, Gene H; VAN LOAN, Charles F. **Matrix computations**. [S.l.]: JHU press, 2013. Citado na p. 83.

HADI, Anas A; MOHAMED, Ali W; JAMBI, Kamal M. LSHADE-SPA memetic framework for solving large-scale optimization problems. **Complex & Intelligent Systems**, Springer, v. 5, n. 1, p. 25–40, 2019. Citado nas pp. 31, 36, 42, 94.

HIBA, Hanan; EL-ABD, Mohammed; RAHNAMAYAN, Shahryar. Improving SHADE with center-based mutation for large-scale optimization. *In: IEEE. 2019 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2019. P. 1533–1540. Citado na p. 31.

HIBA, Hanan; IBRAHIM, Amin; RAHNAMAYAN, Shahryar. Large-scale optimization using center-based differential evolution with dynamic mutation scheme. *In: IEEE. 2019 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2019. P. 3189–3196. Citado nas pp. 31, 39.

HIBA, Hanan; MAHDAVI, Sedigheh; RAHNAMAYAN, Shahryar. Differential evolution with center-based mutation for large-scale optimization. *In: IEEE. 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.: s.n.], 2017. P. 1–8. Citado nas pp. 31, 39.

HIBA, Hanan; RAHNAMAYAN, Shahryar *et al.* A comprehensive investigation on novel center-based sampling for large-scale global optimization. **Swarm and Evolutionary Computation**, Elsevier, v. 73, p. 101105, 2022. Citado nas pp. 30, 39.

JABEEN, Hajira; JALIL, Zunera; BAIG, Abdul Rauf. Opposition based initialization in particle swarm optimization (O-PSO). *In: PROCEEDINGS of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. [S.l.: s.n.], 2009. P. 2047–2052. Citado na p. 49.

JAFARI, Sajad *et al.* Effect of epistasis on the performance of genetic algorithms. **Journal of Zhejiang University-SCIENCE A**, Springer, v. 20, n. 2, p. 109–116, 2019. Citado na p. 19.

JAMIL, Momin; YANG, Xin-She. A literature survey of benchmark functions for global optimisation problems. **International Journal of Mathematical Modelling and Numerical Optimisation**, Inderscience Publishers Ltd, v. 4, n. 2, p. 150–194, 2013. Citado na p. 19.

JIAN, Jun-Rong; CHEN, Zong-Gan *et al.* Region Encoding Helps Evolutionary Computation Evolve Faster: A New Solution Encoding Scheme in Particle Swarm for Large-Scale Optimization. **IEEE Transactions on Evolutionary Computation**, IEEE, 2021. Citado nas pp. 20, 30, 38, 42.

JIAN, Jun-Rong; ZHAN, Zhi-Hui; ZHANG, Jun. Large-scale evolutionary optimization: a survey and experimental comparative study. **International Journal of Machine Learning and Cybernetics**, Springer, v. 11, n. 3, p. 729–745, 2020. Citado nas pp. 32, 37.

KABÁN, Ata; BOOTKRAJANG, Jakramate; DURRANT, Robert J. Toward large-scale continuous EDA: A random matrix theory perspective. **Evolutionary computation**, MIT Press, v. 24, n. 2, p. 255–291, 2016. Citado nas pp. 31, 39.

KARANAM, Santoshachandra Rao; SRINIVAS, Y; CHAKRAVARTY, S. A statistical model approach based on the Gaussian Mixture Model for the diagnosis and classification of bone fractures. **International Journal of Healthcare Management**, Taylor & Francis, p. 1–12, 2023. Citado na p. 61.

KAZIMIPOUR, Borhan; LI, Xiaodong; QIN, A Kai. Effects of population initialization on differential evolution for large scale optimization. *In: IEEE. 2014 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2014. P. 2404–2411. Citado nas pp. 31, 39, 77.

- KAZIMIPOUR, Borhan; OMIDVAR, Mohammad Nabi *et al.* Bandit-based cooperative coevolution for tackling contribution imbalance in large-scale optimization problems. **Applied Soft Computing**, Elsevier, v. 76, p. 265–281, 2019. Citado nas pp. 30, 35.
- KESHK, Marwa; SINGH, Hemant; ABBASS, Hussein. Automatic estimation of differential evolution parameters using Hidden Markov Models. **Evolutionary Intelligence**, Springer, v. 10, p. 77–93, 2018. Citado na p. 45.
- KIM, Kyung Soo; CHOI, Yong Suk. An efficient variable interdependency-identification and decomposition by minimizing redundant computations for large-scale global optimization. **Information Sciences**, Elsevier, v. 513, p. 289–323, 2020. Citado na p. 19.
- KOCER, Havva Gul; UYMAZ, Sait Ali. A Modified Artificial Algae Algorithm For Large Scale Global Optimization Problems. **International Journal of Intelligent Systems and Applications in Engineering**, v. 6, n. 4, p. 306–310, 2018. Citado na p. 20.
- KOÇER, Havva Gül; UYMAZ, Sait Ali. A novel local search method for LSGO with golden ratio and dynamic search step. **Soft Computing**, Springer, v. 25, n. 3, p. 2115–2130, 2021. Citado nas pp. 20, 30, 36.
- KOROŠEC, Peter; ŠILC, Jurij. The continuous differential ant-stigmergy algorithm applied on real-parameter single objective optimization problems. *In: IEEE. 2013 IEEE Congress on Evolutionary Computation. [S.l.: s.n.], 2013. P. 1658–1663. Citado nas pp. 31, 37.*
- KOZA, John R *et al.* Evolution of subsumption using genetic programming. *In: MIT PRESS CAMBRIDGE, MA, USA. PROCEEDINGS of the first European conference on artificial life. [S.l.: s.n.], 1992. P. 110–119. Citado na p. 44.*
- KRAMER, Oliver. Evolutionary self-adaptation: a survey of operators and strategy parameters. **Evolutionary Intelligence**, Springer, v. 3, p. 51–65, 2010. Citado nas pp. 45, 46.
- LANGE, Kenneth; LANGE, Kenneth. Singular value decomposition. **Numerical analysis for statisticians**, Springer, p. 129–142, 2010. Citado nas pp. 54–56.
- LATORRE, Antonio; MUELAS, Santiago; PENA, Jose-Maria. Large scale global optimization: Experimental results with MOS-based hybrid algorithms. *In: IEEE. 2013*

IEEE congress on evolutionary computation. [S.l.: s.n.], 2013. P. 2742–2749. Citado nas pp. 31, 37.

LATORRE, Antonio; MUELAS, Santiago; PENA, Jose-Maria. Multiple offspring sampling in large scale global optimization. *In: IEEE. 2012 IEEE Congress on Evolutionary Computation.* [S.l.: s.n.], 2012. P. 1–8. Citado nas pp. 31, 37.

LI, Dongyang; GUO, Weian *et al.* An adaptive particle swarm optimizer with decoupled exploration and exploitation for large scale optimization. **Swarm and Evolutionary Computation**, Elsevier, v. 60, p. 100789, 2021. Citado nas pp. 19–21, 30.

LI, Dongyang; WANG, Lei *et al.* A particle swarm optimizer with dynamic balance of convergence and diversity for large-scale optimization. **Applied Soft Computing**, Elsevier, v. 132, p. 109852, 2023. Citado nas pp. 19, 30, 38.

LI, Lin *et al.* Cooperative coevolution for large-scale global optimization based on fuzzy decomposition. **Soft Computing**, Springer, v. 25, n. 5, p. 3593–3608, 2021. Citado nas pp. 19–21, 30, 34.

LI, Xiaodong *et al.* Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. **gene**, v. 7, n. 33, p. 8, 2013. Citado nas pp. 29, 91, 93.

LIU, Haiyan; CHENG, Yuan *et al.* A space-reduction based three-phase approach for large-scale optimization. **Applied Soft Computing**, Elsevier, p. 110517, 2023. Citado na p. 19.

LIU, Haiyan; GUAN, Shiwei *et al.* Cooperative co-evolution with formula based grouping and CMA for large scale optimization. *In: IEEE. 2015 11th International Conference on Computational Intelligence and Security (CIS).* [S.l.: s.n.], 2015. P. 282–285. Citado nas pp. 31, 34.

LIU, Haiyan; WANG, Yuping *et al.* A new grouping strategy-based hybrid algorithm for large scale global optimization problems. *In: PROCEEDINGS of the Genetic and Evolutionary Computation Conference Companion.* [S.l.: s.n.], 2017. P. 171–172. Citado nas pp. 31, 34.

LIU, Jinpeng; TANG, Ke. Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution. *In: SPRINGER. INTERNATIONAL Conference on*

Intelligent Data Engineering and Automated Learning. [S.l.: s.n.], 2013. P. 350–357. Citado nas pp. 31, 42, 94.

LUAN, Xiaojie *et al.* Decomposition and distributed optimization of real-time traffic management for large-scale railway networks. **Transportation Research Part B: Methodological**, Elsevier, v. 141, p. 72–97, 2020. Citado na p. 19.

MA, Yongjie; BAI, Yulong. A multi-population differential evolution with best-random mutation strategy for large-scale global optimization. **Applied Intelligence**, Springer, v. 50, n. 5, p. 1510–1526, 2020. Citado nas pp. 30, 39.

MAHDAVI, Sedigheh; RAHNAMAYAN, Shahryar; DEB, Kalyanmoy. Center-based initialization of cooperative co-evolutionary algorithm for large-scale optimization. *In: IEEE. 2016 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2016. P. 3557–3565. Citado nas pp. 31, 33.

MAHDAVI, Sedigheh; RAHNAMAYAN, Shahryar; DEB, Kalyanmoy. Opposition based learning: A literature review. **Swarm and evolutionary computation**, Elsevier, v. 39, p. 1–23, 2018. Citado nas pp. 49, 51–53, 77, 78.

MAHDAVI, Sedigheh; RAHNAMAYAN, Shahryar; SHIRI, Mohammad Ebrahim. Incremental cooperative coevolution for large-scale global optimization. **Soft Computing**, Springer, v. 22, n. 6, p. 2045–2064, 2018. Citado nas pp. 31, 35.

MAHDAVI, Sedigheh; RAHNAMAYAN, Shahryar; SHIRI, Mohammad Ebrahim. Multilevel framework for large-scale global optimization. **Soft Computing**, Springer, v. 21, n. 14, p. 4111–4140, 2017. Citado nas pp. 31, 35.

MARCELINO, Carolina *et al.* Applying C-DEEPSO to solve large scale global optimization problems. *In: IEEE Congress on Evolutionary Computation*. [S.l.: s.n.], 2018. P. 1–6. Citado nas pp. 31, 42, 94.

MAUČEC, Mirjam Sepesy; BREST, Janez. A review of the recent use of Differential Evolution for Large-Scale Global Optimization: An analysis of selected algorithms on the CEC 2013 LSGO benchmark suite. **Swarm and Evolutionary Computation**, Elsevier, v. 50, p. 100428, 2019. Citado nas pp. 20, 45.

MAUČEC, Mirjam Sepesy; BREST, Janez; BOŠKOVIĆ, Borko *et al.* Improved differential evolution for large-scale black-box optimization. **IEEE Access**, IEEE, v. 6, p. 29516–29531, 2018. Citado nas pp. 31, 39.

MCLACHLAN, Geoffrey J; LEE, Sharon X; RATHNAYAKE, Suren I. Finite mixture models. **Annual review of statistics and its application**, Annual Reviews, v. 6, p. 355–378, 2019. Citado na p. 58.

MITCHELL, Melanie. **An introduction to genetic algorithms**. [S.l.]: MIT press, 1998. Citado na p. 44.

MOHAPATRA, Prabhujit; DAS, Kedar Nath; ROY, Santanu. A modified competitive swarm optimizer for large scale optimization problems. **Applied Soft Computing**, Elsevier, v. 59, p. 340–362, 2017. Citado nas pp. 31, 38.

MOLINA, Daniel; HERRERA, Francisco. Iterative hybridization of DE with local search for the CEC'2015 special session on large scale global optimization. *In: IEEE Congress on Evolutionary Computation*. [S.l.: s.n.], 2015. P. 1974–1978. Citado nas pp. 31, 42, 94.

MOLINA, Daniel; LATORRE, Antonio. Toolkit for the automatic comparison of optimizers: comparing large-scale global optimizers made easy. *In: IEEE Congress on Evolutionary Computation*. [S.l.: s.n.], 2018. P. 1–8. Citado nas pp. 93, 94.

MOLINA, Daniel; LATORRE, Antonio; HERRERA, Francisco. SHADE with iterative local search for large-scale global optimization. *In: IEEE. 2018 IEEE congress on evolutionary computation (CEC)*. [S.l.: s.n.], 2018. P. 1–8. Citado nas pp. 20, 31, 39, 42, 83, 86, 94.

MOLINA, Daniel; NESTERENKO, Arthur R; LATORRE, Antonio. Comparing Large-Scale Global Optimization Competition winners in a real-world problem. *In: IEEE. 2019 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2019. P. 359–365. Citado nas pp. 20, 45, 95.

MORALES, José Luis; NOCEDAL, Jorge. Remark on “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization”. **ACM Transactions on Mathematical Software (TOMS)**, ACM New York, NY, USA, v. 38, n. 1, p. 1–4, 2011. Citado nas pp. 43, 88.

MOUSAVIRAD, Seyed Jalaleddin; RAHNAMAYAN, Shahryar. Evolving feedforward neural networks using a quasi-opposition-based differential evolution for data classification. *In: IEEE. 2020 IEEE symposium series on computational intelligence (SSCI)*. [S.l.: s.n.], 2020. P. 2320–2326. Citado na p. 19.

NEIVA, Frâncila Weidt; SILVA, RL d S da. **Revisao sistemática da literatura em ciência da computação um guia prático**. [S.l.], 2016. Citado na p. 28.

OMIDVAR, Mohammad Nabi; LI, Xiaodong. Evolutionary large-scale global optimization: an introduction. *In: PROCEEDINGS of the Genetic and Evolutionary Computation Conference Companion*. [S.l.: s.n.], 2017. P. 807–827. Citado na p. 19.

OMIDVAR, Mohammad Nabi; LI, Xiaodong; TANG, Ke. Designing benchmark problems for large-scale continuous optimization. **Information Sciences**, Elsevier, v. 316, p. 419–436, 2015. Citado na p. 94.

OMIDVAR, Mohammad Nabi; LI, Xiaodong; YAO, Xin. A review of population-based metaheuristics for large-scale black-box global optimization—Part I. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 26, n. 5, p. 802–822, 2021. Citado na p. 44.

OMIDVAR, Mohammad Nabi; LI, Xiaodong; YAO, Xin. A review of population-based metaheuristics for large-scale black-box global optimization: Part B. **IEEE Transactions on Evolutionary Computation**, p. 1–1, 2021. DOI: 10.1109/TEVC.2021.3130835. Citado na p. 93.

OMIDVAR, Mohammad Nabi; LI, Xiaodong; YAO, Xin. Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms. *In: PROCEEDINGS of the 13th annual conference on Genetic and evolutionary computation*. [S.l.: s.n.], 2011. P. 1115–1122. Citado na p. 35.

OSTERTAGOVA, Eva; OSTERTAG, Oskar; KOVÁČ, Jozef. Methodology and application of the Kruskal-Wallis test. *In: TRANS TECH PUBL. APPLIED Mechanics and Materials*. [S.l.: s.n.], 2014. P. 115–120. Citado nas pp. 92, 104.

PACHECO-DEL-MORAL, Oscar; COELLO, Carlos A Coello. A SHADE-Based Algorithm for Large Scale Global Optimization. *In: SPRINGER. INTERNATIONAL Conference on Parallel Problem Solving from Nature*. [S.l.: s.n.], 2020. P. 650–663. Citado nas pp. 20, 30, 42, 94.

PATEL, Rahila; RAGHUWANSHI, MM; MALIK, LG. Decomposition based multi-objective genetic algorithm (DMOGA) with opposition based learning. *In: IEEE. 2012 Fourth International Conference on Computational Intelligence and Communication Networks*. [S.l.: s.n.], 2012. P. 605–610. Citado na p. 49.

PROTTER, Murray H; CHARLES JR, B *et al.* **A first course in real analysis**. [S.l.]: Springer Science & Business Media, 2012. Citado na p. 66.

QIN, A Kai; SUGANTHAN, Ponnuthurai N. Self-adaptive differential evolution algorithm for numerical optimization. *In: IEEE. 2005 IEEE congress on evolutionary computation*. [S.l.: s.n.], 2005. P. 1785–1791. Citado na p. 46.

REN, Zhigang *et al.* An eigenspace divide-and-conquer approach for large-scale optimization. **Applied Soft Computing**, Elsevier, v. 99, p. 106911, 2021. Citado nas pp. 20, 21, 30, 36.

RODRIGUES, Douglas *et al.* Opposition-Based Jellyfish Search for Feature Selection. *In: IEEE. 2023 30th International Conference on Systems, Signals and Image Processing (IWSSIP)*. [S.l.: s.n.], 2023. P. 1–5. Citado nas pp. 30, 41, 42, 94.

SANYANG, Momodou L; KABAN, Ata. Large-Scale Estimation of Distribution Algorithms with Adaptive Heavy Tailed Random Projection Ensembles. **Journal of Computer Science and Technology**, Springer, v. 34, n. 6, p. 1241–1257, 2019. Citado nas pp. 31, 39.

SCHOEN, Fabio; TIGLI, Luca. Efficient large scale global optimization through clustering-based population methods. **Computers & Operations Research**, Elsevier, v. 127, p. 105165, 2021. Citado nas pp. 20, 21, 30, 40, 42.

SEGREDO, Eduardo *et al.* On the comparison of initialisation strategies in differential evolution for large scale optimisation. **Optimization Letters**, Springer, v. 12, n. 1, p. 221–234, 2018. Citado nas pp. 31, 39.

SONG, Chunxiang; WANG, Xiaojun; CAI, Yiqiao. Evolutionary Multitasking with Scale-Adaptive Subspace for Large-Scale Optimization. *In: IEEE. 2022 IEEE 2nd International Conference on Software Engineering and Artificial Intelligence (SEAI)*. [S.l.: s.n.], 2022. P. 160–165. Citado nas pp. 30, 41, 42.

SOPOV, Evgenii; VAKHNIN, Alexey. Investigating the effects of population size and the number of subcomponents on the performance of SHADE algorithm with random adaptive grouping for LSGO problems. *In: PROCEEDINGS of the Genetic and Evolutionary Computation Conference Companion*. [S.l.: s.n.], 2019. P. 269–270. Citado nas pp. 31, 33.

SORENSEN, Harold Wayne. Parameter estimation: principles and problems. **(No Title)**, 1980. Citado na p. 60.

SORIA-ALCARAZ JORGE, A *et al.* Comparison of metaheuristic algorithms with a methodology of design for the evaluation of hard constraints over the course timetabling problem. *In: RECENT Advances on Hybrid Intelligent Systems*. [S.l.]: Springer, 2013. P. 289–302. Citado na p. 104.

STORN, Rainer; PRICE, Kenneth. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. **Journal of global optimization**, Springer, v. 11, p. 341–359, 1997. Citado nas pp. 44, 45, 47, 48.

SUN, Mai *et al.* Large-scale expensive optimization with a switching strategy. **Complex System Modeling and Simulation**, TUP, v. 2, n. 3, p. 253–263, 2022. Citado nas pp. 30, 34.

SUN, Yuan *et al.* Decomposition for large-scale optimization problems with overlapping components. *In: IEEE. 2019 IEEE congress on evolutionary computation (CEC)*. [S.l.: s.n.], 2019. P. 326–333. Citado nas pp. 31, 35.

TABERNIK, Domen; SKOČAJ, Danijel. Deep learning for large-scale traffic-sign detection and recognition. **IEEE transactions on intelligent transportation systems**, IEEE, v. 21, n. 4, p. 1427–1440, 2019. Citado na p. 19.

TAMANG, Jharna *et al.* Dynamical properties of ion-acoustic waves in space plasma and its application to image encryption. **IEEE Access**, IEEE, v. 9, p. 18762–18782, 2021. Citado na p. 19.

TANG, Ke; LI, Xiaodong *et al.* **Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization**. [S.l.], 2010.

Disponível em:

<<http://www.it-weise.de/documents/files/TLSYW2009BFFTCSSACOLSGO.pdf>>. Citado na p. 91.

TANG, Ke; YÁO, Xin *et al.* Benchmark functions for the CEC'2008 special session and competition on large scale global optimization. **Nature inspired computation and applications laboratory, USTC, China**, v. 24, p. 1–18, 2007. Citado na p. 91.

TIZHOOSH, H.R. Opposition-Based Learning: A New Scheme for Machine Intelligence. *In: INTERNATIONAL Conference on Computational Intelligence for*

Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). [S.l.: s.n.], 2005. P. 695–701. DOI: 10.1109/CIMCA.2005.1631345. Citado na p. 49.

TIZHOOSH, Hamid R. Opposite Fuzzy Sets with Applications in Image Processing. *In*: CITESEER. IFSA/EUSFLAT Conf. [S.l.: s.n.], 2009. P. 36–41. Citado na p. 49.

TIZHOOSH, Hamid R. Opposition-based learning: a new scheme for machine intelligence. *In*: IEEE. INTERNATIONAL conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06). [S.l.: s.n.], 2005. P. 695–701. Citado na p. 48.

TIZHOOSH, Hamid R. Reinforcement learning based on actions and opposite actions. *In*: INTERNATIONAL conference on artificial intelligence and machine learning. [S.l.: s.n.], 2005. Citado na p. 49.

TOUGAS, Jane E; SPITERI, Raymond J. Updating the partial singular value decomposition in latent semantic indexing. **Computational Statistics & Data Analysis**, Elsevier, v. 52, n. 1, p. 174–183, 2007. Citado nas pp. 68, 80.

TSENG, Lin-Yu; CHEN, Chun. Multiple trajectory search for large scale global optimization. *In*: IEEE. 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). [S.l.: s.n.], 2008. P. 3052–3059. Citado nas pp. 43, 88.

TYAGI, Hemant; CEVHER, Volkan. Learning non-parametric basis independent models from point queries via low-rank methods. **Applied and Computational Harmonic Analysis**, Elsevier, v. 37, n. 3, p. 389–412, 2014. Citado na p. 67.

VARELAS, Konstantinos *et al.* Benchmarking large-scale continuous optimizers: The bbob-largescale testbed, a COCO software guide and beyond. **Applied Soft Computing**, Elsevier, v. 97, p. 106737, 2020. Citado na p. 29.

VENTRESCA, Mario; TIZHOOSH, Hamid R. Improving the convergence of backpropagation by opposite transfer functions. *In*: IEEE. THE 2006 IEEE International Joint Conference on Neural Network Proceedings. [S.l.: s.n.], 2006. P. 4777–4784. Citado na p. 49.

VENTRESCA, Mario; TIZHOOSH, Hamid R. Simulated annealing with opposite neighbors. *In: IEEE. 2007 IEEE Symposium on Foundations of Computational Intelligence*. [S.l.: s.n.], 2007. P. 186–192. Citado na p. 49.

VYBIRAL, Jan. Learning Functions of Few Arbitrary Linear Parameters in High Dimensions. Citeseer. Citado na p. 66.

WANG, Feng; WANG, Xujie; SUN, Shilei. A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. **Information Sciences**, Elsevier, v. 602, p. 298–312, 2022. Citado na p. 38.

WANG, Xiang; JIN, Xiaoming. Understanding and enhancing the folding-in method in latent semantic indexing. *In: SPRINGER. INTERNATIONAL Conference on Database and Expert Systems Applications*. [S.l.: s.n.], 2006. P. 104–113. Citado nas pp. 68, 80.

WANG, Yuping; LIU, Haiyan *et al.* Cooperative coevolution with formula-based variable grouping for large-scale global optimization. **Evolutionary computation**, MIT Press, v. 26, n. 4, p. 569–596, 2018. Citado nas pp. 31, 34.

WEI, Fei; WANG, Yuping; HUO, Yuanliang. Smoothing and auxiliary functions based cooperative coevolution for global optimization. *In: IEEE. 2013 IEEE Congress on Evolutionary Computation*. [S.l.: s.n.], 2013. P. 2736–2741. Citado nas pp. 31, 32.

WEI, Fei; WANG, Yuping; ZONG, Tingting. A novel cooperative coevolution for large scale global optimization. *In: IEEE. 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.: s.n.], 2014. P. 738–741. Citado nas pp. 31, 33.

WEI, Fei; WANG, Yuping; ZONG, Tingting. Variable grouping based differential evolution using an auxiliary function for large scale global optimization. *In: IEEE. 2014 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2014. P. 1293–1298. Citado nas pp. 31, 34.

WOLPERT, David H; MACREADY, William G. No free lunch theorems for optimization. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 1, n. 1, p. 67–82, 1997. Citado na p. 112.

XUE, Xiaoming *et al.* A topology-based single-pool decomposition framework for large-scale global optimization. **Applied Soft Computing**, Elsevier, v. 92, p. 106295, 2020. Citado nas pp. 20, 30, 36.

YANG, Ming *et al.* CCFR2: A more efficient cooperative co-evolutionary framework for large-scale global optimization. **Information Sciences**, Elsevier, v. 512, p. 64–79, 2020. Citado nas pp. 30, 35.

YU, Dong *et al.* Gaussian mixture models. **Automatic Speech Recognition: A Deep Learning Approach**, Springer, p. 13–21, 2015. Citado na p. 61.

ZAHARIE, Daniela. Control of population diversity and adaptation in differential evolution algorithms. *In: PROC. of MENDEL*. [S.l.: s.n.], 2003. P. 41–46. Citado na p. 46.

ZHANG, Kaibo; LI, Bin. Cooperative coevolution with global search for large scale global optimization. *In: IEEE. 2012 IEEE Congress on Evolutionary Computation*. [S.l.: s.n.], 2012. P. 1–7. Citado na p. 32.

ZHANG, Tianya; JIN, Peter J; GE, Yi. Multimodal Gaussian Mixture Model for Realtime Roadside LiDAR Object Detection. **arXiv preprint arXiv:2204.09804**, 2022. Citado na p. 61.

ZHANG, Xin; ZHAN, Zhi-Hui; ZHANG, Jun. Adaptive Population Differential Evolution with Dual Control Strategy for Large-Scale Global optimization Problems. *In: IEEE. 2020 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2020. P. 1–7. Citado nas pp. 20, 30, 39.

ZHANG, Xin-Yuan; GONG, Yue-Jiao *et al.* Dynamic cooperative coevolution for large scale optimization. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 23, n. 6, p. 935–948, 2019. Citado nas pp. 30, 34.

ZHAO, Dong *et al.* Opposition-based ant colony optimization with all-dimension neighborhood search for engineering design. **Journal of Computational Design and Engineering**, Oxford University Press, v. 9, n. 3, p. 1007–1044, 2022. Citado na p. 49.