



Pós-Graduação em Ciência da Computação

Mário Gomes de Melo

Adaptação Local da Matriz de Covariância Guiada por Mecanismos de Exploração em Estratégias de Evolução



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
<http://cin.ufpe.br/~posgraduacao>

Recife
2019

Mário Gomes de Melo

**Adaptação Local da Matriz de Covariância Guiada por Mecanismos de Exploração
em Estratégias de Evolução**

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de Concentração: Inteligência Computacional

Orientador: Prof. Dr. Aluizio Fausto Ribeiro Araújo

Recife
2019

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

M528a Melo, Mário Gomes de
Adaptação local da matriz de covariância guiada por mecanismos de exploração em estratégias de evolução / Mário Gomes de Melo. – 2019.
116 f.: il., fig., tab.

Orientador: Aluizio Fausto Ribeiro Araújo.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2019.
Inclui referências.

1. Inteligência computacional. 2. Otimização. 3. Computação evolucionária.
I. Araújo, Aluizio Fausto Ribeiro (orientador). II. Título.

006.3

CDD (23. ed.)

UFPE- MEI 2019-065

Mário Gomes de Melo

"Adaptação Local da Matriz de Covariância Guiada por Mecanismos de Exploração em Estratégias de Evolução"

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 26/02/2019.

BANCA EXAMINADORA

Prof. Dr. Abel Guilhermino da Silva Filho
Centro de Informática / UFPE

Prof. Dr. Carmelo Jose Albanez Bastos Filho
Escola Politécnica de Pernambuco / UPE

Prof. Dr. Aluizio Fausto Ribeiro Araújo
Centro de Informática / UFPE
(Orientador)

Dedico este trabalho a minha família, especialmente a memória da minha avó Alice, aos meus amigos, a todos os professores que passaram na minha vida e dividiram um pouco de seus conhecimentos comigo, e não menos importante: a ciência, pois tudo isso ocorre devido ao desejo de contribuir cada vez mais para seu avanço.

AGRADECIMENTOS

Agradeço a Deus por ter me dado forças para seguir adiante, aos meus familiares que sempre estiveram ao meu lado, me auxiliando nesta jornada, aos meus professores e aos meus colegas do meio acadêmico e profissional que nunca mediram esforços para prestar ajuda, a fim de que eu pudesse chegar até aqui.

"Na medida em que as proposições da matemática se referem à realidade, elas não têm certeza, e na medida em que têm certeza, não se referem à realidade."
(EINSTEIN, 1921, pp. 3-4, tradução nossa)¹

¹ *"Insofern sich die Sätze der Mathematik auf die Wirklichkeit beziehen, sind sie nicht sicher, und insofern sie sicher sind, beziehen sie sich nicht auf die Wirklichkeit."*

RESUMO

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) é o Estado-da-Arte, em computação evolucionária, para otimização de funções mono-objetivo no domínio dos números reais. Contudo, apresenta algumas limitações, como a dependência do ponto de inicialização e do tamanho da população, que podem conduzi-lo à convergência prematura, principalmente em funções multimodais. Portanto, agregar ao CMA-ES um mecanismo que possibilite explorar o espaço de busca, à procura de regiões mais promissoras, apresenta-se como uma relevante alternativa para mitigar estas limitações. Ao longo dos anos, foram desenvolvidos modelos baseados no CMA-ES que utilizam algum processo de exploração para evitar tais comportamentos, desde aqueles que aplicam reinicialização do processo evolucionário aos que empregam outro algoritmo com características exploratórias, como o *Differential Evolution* (DE), obtendo-se portanto um algoritmo memético. Com base nas evidências apresentadas, foi desenvolvido neste trabalho o *Local CMA-ES Guided by Differential Evolution* (L-CMA-DE), com o objetivo de aproveitar a sinergia entre a reinicialização e algoritmos meméticos, ao combinar CMA-ES e DE para produzir melhores soluções sem aumentar o número de avaliações de função. Logo, o algoritmo proposto é fundamentado em dois processos adaptativos: exploração em subespaços independentes, através do CMA-ES, guiada por uma exploração global, realizada pelo DE. Os experimentos realizados sugerem que o L-CMA-DE apresenta melhores resultados quando comparado ao CMA-ES e ao DE, aplicados isoladamente. A nova proposta também apresenta uma melhor performance quando comparada a algoritmos Estado-da-Arte, em funções de maior complexidade, sem prejuízo na quantidade de avaliações de função.

Palavras-chaves: Inteligência Computacional. Otimização. Computação Evolucionária. Adaptação da Matriz de Covariância. Evolução Diferencial.

ABSTRACT

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is the State-of-Art, in evolutionary algorithms to optimizing mono-objective functions in the real-number domain. However, CMA-ES presents some limitations, such as the dependency of the initialization point and population size, which can lead to premature convergence, mainly in multimodal functions. Therefore, to insert in CMA-ES a mechanism that makes it possible to explore the search space, looking for more promising regions, is a relevant alternative to mitigate the mentioned limitations. Along the years, different researchers proposed versions of CMA-ES that improved the exploration, often considering re-start procedures or another algorithm with exploratory characteristics, such as *Differential Evolution* (DE), obtaining a memetic algorithm. Based on the evidence presented, we propose the *Local CMA-ES Guided by Differential Evolution* (L-CMA-DE) with the aim of taking advantage of the synergy between restart and memetic algorithms combining CMA-ES and DE to produce better solutions without adding to the number of function evaluations. Therefore, the L-CMA-DE is based on two adaptive processes: independent subspaces exploitation, through CMA-ES, guided by a global exploration, performed by DE. The experiments suggest that the L-CMA-DE presents better results when compared to CMA-ES and DE, individually applied. The new proposal also presents a better performance when compared to State-of-Art algorithms, in functions of high complexity, without adding to the number of function evaluations.

Keywords: Computational Intelligence. Optimization. Evolutionary Computing. Covariance Matrix Adaptation. Differential Evolution.

LISTA DE FIGURAS

Figura 1 – Arcabouço de um problema de simulação	18
Figura 2 – Arcabouço de um problema de modelagem	18
Figura 3 – Arcabouço de um problema de otimização	19
Figura 4 – Arcabouço geral de um algoritmo evolucionário	19
Figura 5 – Efeito da adaptação da matriz de covariância	22
Figura 6 – Configurações de trajetória de evolução	27
Figura 7 – Evolução da população do CMA-ES	30
Figura 8 – Dois pontos detectados como um par 2-modal	39
Figura 9 – Dois pontos detectados como um par 1-modal	40
Figura 10 – Representação da sensibilidade ao parâmetro <i>percentFESCMAES</i>	73
Figura 11 – Representação da sensibilidade do parâmetro <i>stagCMAES</i>	74
Figura 12 – Representação da sensibilidade ao parâmetro <i>stagDE</i>	75
Figura 13 – Representação da sensibilidade ao parâmetro <i>sparsityLevel</i> (comportamento Mínimo)	75
Figura 14 – Representação da sensibilidade ao parâmetro <i>sparsityLevel</i> (comportamento Neutro)	76
Figura 15 – Representação da sensibilidade ao parâmetro <i>refValue</i> (comportamento Neutro)	76
Figura 16 – Representação da sensibilidade ao parâmetro <i>maxGenDE</i>	77
Figura 17 – Comportamento quanto à variação do parâmetro <i>percentFESCMAES</i>	78
Figura 18 – Comportamento quanto à variação do parâmetro <i>stagDE</i>	78
Figura 19 – Comportamento quanto à variação do parâmetro <i>sparsityLevel</i>	79
Figura 20 – Representação em 3- <i>D</i> dos tipos de funções (espaço de busca em 2- <i>D</i>)	83

LISTA DE TABELAS

Tabela 1 – Esparsidade da matriz de covariância ao longo das gerações (Função Esfera)	48
Tabela 2 – Esparsidade da matriz de covariância ao longo das gerações (Função 22 da CEC-2013)	50
Tabela 3 – <i>Mann-Whitney-Wilcoxon Test</i> (CMA-ES × CMA-ES - diversidade com os μ melhores)	52
Tabela 4 – <i>Mann-Whitney-Wilcoxon Test</i> (CMA-ES × CMA-ES - diversidade com os μ melhores - aptidão do processo alternativo sendo a distância para o centroide)	53
Tabela 5 – <i>Mann-Whitney-Wilcoxon Test</i> (CMA-ES × CMA-ES - aplicando diversidade mútua entre a população principal e a alternativa)	55
Tabela 6 – <i>Mann-Whitney-Wilcoxon Test</i> (CMA-ES × CMAESDivRand_200)	57
Tabela 7 – <i>Mann-Whitney-Wilcoxon Test</i> (CMAES_DivRandDERepMean × CMAESDivRand_200)	59
Tabela 8 – Convergência com a utilização do método da Roleta para a escolha do espaço de busca (2-D)	62
Tabela 9 – CMA-ES versus CMA-ES com compressão do espaço de busca de [-100,100] para [-25,25] (2-D)	64
Tabela 10 – <i>Mann-Whitney-Wilcoxon Test</i> (CMA-ES × CMA-ES-PSS) (2-D)	66
Tabela 11 – Comportamento das funções quanto à variação de <i>percentFESCMAES</i>	73
Tabela 12 – Comportamento das funções quanto à variação de <i>stagCMAES</i>	74
Tabela 13 – Comportamento das funções quanto à variação de <i>stagDE</i>	74
Tabela 14 – Comportamento das funções quanto à variação de <i>maxGenDE</i>	77
Tabela 15 – Valores paramétricos utilizados para funções da CEC-2017	79
Tabela 16 – Valores paramétricos utilizados para funções da CEC-2013	80
Tabela 17 – Funções da CEC-2013	81
Tabela 18 – Funções da CEC-2017	82
Tabela 19 – Testes com o algoritmo proposto e o CMA-ES (CEC-2013 / 10-D)	89
Tabela 20 – Testes com o algoritmo proposto e o DE (CEC-2013 / 10-D)	90
Tabela 21 – Testes com o algoritmo proposto e o CMA-ES-RIS (CEC-2013 / 10-D)	91
Tabela 22 – Testes com o algoritmo proposto e o IPOPOP-CMA-ES (CEC-2013 / 10-D)	92
Tabela 23 – Testes com o algoritmo proposto e o NBIPOP-aCMA-ES (CEC-2013 / 10-D)	93
Tabela 24 – Testes com o algoritmo proposto e o CMA-ES (CEC-2013 / Funções Compostas / 10-D)	94

Tabela 25 – Testes com o algoritmo proposto e o DE (CEC-2013 / Funções Compostas / 10- D)	95
Tabela 26 – Testes com o algoritmo proposto e o CMA-ES-RIS (CEC-2013 / Funções Compostas / 10- D)	96
Tabela 27 – Testes com o algoritmo proposto e o IPOP-CMA-ES (CEC-2013 / Funções Compostas / 10- D)	97
Tabela 28 – Testes com o algoritmo proposto e o NBIPOP-aCMA-ES (CEC-2013 / Funções Compostas / 10- D)	98
Tabela 29 – Testes com o algoritmo proposto e o CMA-ES (CEC-2017 / 10- D)	99
Tabela 30 – Testes com o algoritmo proposto e o DE (CEC-2017 / 10- D)	100
Tabela 31 – Testes com o algoritmo proposto e o RB-IPOP-CMA-ES (CEC-2017 / 10- D)	101
Tabela 32 – Testes com o algoritmo proposto e o LSHADE-SPACMA (CEC-2017 / 10- D)	102
Tabela 33 – Testes com o algoritmo proposto e o DE (CEC-2017 / Funções Compostas / 10- D)	103
Tabela 34 – Testes com o algoritmo proposto e o RB-IPOP-CMA-ES (CEC-2017 / Funções Compostas / 10- D)	104
Tabela 35 – Testes com o algoritmo proposto e o LSHADE-SPACMA (CEC-2017 / Funções Compostas / 10- D)	105
Tabela 36 – Análise Geral (28 funções da CEC-2013 / 10- D)	106
Tabela 37 – Análise Geral (5 funções unimodais da CEC-2013 / 10- D)	106
Tabela 38 – Análise Geral (15 funções multimodais básicas da CEC-2013 / 10- D)	106
Tabela 39 – Análise Geral (8 funções compostas da CEC-2013 / 10- D)	106
Tabela 40 – Análise Geral (29 funções da CEC-2017 / 10- D)	106
Tabela 41 – Análise Geral (2 funções unimodais da CEC-2017 / 10- D)	106
Tabela 42 – Análise Geral (7 funções multimodais simples da CEC-2017 / 10- D)	107
Tabela 43 – Análise Geral (10 funções híbridas da CEC-2017 / 10- D)	107
Tabela 44 – Análise Geral (10 funções compostas da CEC-2017 / 10- D)	107

LISTA DE ABREVIATURAS E SIGLAS

$(\mu/\mu_w, \lambda)$ -aCMA-ES	<i>Weighted Active Covariance Matrix Adaptation Evolution Strategy</i>
<i>Active</i> -CMA-ES	<i>Active Covariance Matrix Adaptation Evolution Strategy</i>
ABS	<i>Attraction Basin Sphere</i>
ABSE	<i>Attraction Basin Sphere Estimation</i>
ABSGA	<i>Attraction Basin Sphere Genetic Algorithm</i>
BIPOP-aCMA-ES	<i>BI-population weighted-aCMA-ES</i>
BIPOP-CMA-ES	<i>BI-population Covariance Matrix Adaptation Evolution Strategy</i>
CEC	<i>IEEE Congress on Evolutionary Computation</i>
CMA-ES-DTBP	<i>Covariance Matrix Adaptation Evolution Strategy with Directed Target to Best Perturbation Scheme</i>
CMA-ES	<i>Covariance Matrix Adaptation Evolution Strategy</i>
CSA	<i>Cumulative Step-size Adaptation</i>
DE	<i>Differential Evolution</i>
DECMSA	<i>Differential Evolution variant with Covariance Matrix Adaptation Evolution Strategy</i>
DTBP	<i>Directed Target to Best Perturbation</i>
EE	<i>Estratégia de Evolução</i>
FEM	<i>Finite Element Method</i>
fGA	<i>forking Genetic Algorithm</i>
GA-SSS-PSO	<i>Genetic Algorithm Search Space Splicing Particle Swarm Optimization</i>
GB	<i>Gigabyte</i>
g-fGA	<i>genotypic forking Genetic Algorithm</i>
GHz	<i>Gigahertz</i>
IPOP-aCMA-ES	<i>restart-(μ_w, λ)-weighted-aCMA-ES with Increasing Population</i>
IPOP-CMA-ES	<i>restart-(μ_w, λ)-CMA-ES with Increasing Population</i>
L-CMA-DE	<i>Local CMA-ES Guided by Differential Evolution</i>

LHS	<i>Latin Hypercube Sampling</i>
MEF	Método dos Elementos Finitos
NBC	<i>Nearest-better Clustering</i>
NBIPOP-aCMA-ES	<i>New BI-population weighted-aCMA-ES with Increasing Population</i>
NIPOP-aCMA-ES	<i>New weighted-aCMA-ES with Increasing Population</i>
NMVO	<i>Niching Variable Mesh Optimization</i>
p-fGA	<i>phenotypic forking Genetic Algorithm</i>
RAM	<i>Random Access Memory</i>

LISTA DE SÍMBOLOS

α	alfa: nível de significância do teste estatístico
\sim	aproximadamente: indica como o modelo probabilístico, construído no CMA-ES, gera um indivíduo
β	beta: espaço de busca original do problema
\mathcal{O}	<i>big O</i> : notação para indicar o pior caso do algoritmo, referente à complexidade de tempo
Δ	Delta: variação de um determinado parâmetro
η	eta: melhores indivíduos de um determinado nicho
ϕ	fi: ponderador para influência da média aritmética das arestas que ligam as soluções
γ	gama: quantidade de indivíduos gerados através de um mecanismo diverso do CMA-ES
Γ	Gama: região contida no espaço de busca original
λ	lambda: quantidade de indivíduos de uma geração
$\lfloor \]$	piso de um determinado valor
μ	mi: quantidade de melhores indivíduos utilizados para guiar o processo evolucionário do CMA-ES
ω	ômega: fator ponderador $\in [0 \dots 1]$
\forall	para todo
\in	pertence
ψ	psi: limiar para remoção de arestas entre soluções
ρ	rô: raio de atuação de um nicho
σ	sigma: passo de mutação
Θ	Teta: sendo $\Theta(\cdot)$ a representação da função degrau
$\lceil \]$	teto de um determinado valor
ζ	zeta: parâmetro que pondera a diferença entre os passos de duas populações consecutivas

LISTA DE ALGORITMOS

Algoritmo 1	–	CMA-ES	30
Algoritmo 2	–	<i>ES Dynamic Niching Algorithm</i>	34
Algoritmo 3	–	<i>Dynamic Niching</i> com $(1, \lambda)$ -CMA-ES: uma única iteração . . .	35
Algoritmo 4	–	<i>Dynamic Peak Identification (DPI)</i>	35
Algoritmo 5	–	$(1, \lambda)$ -CMA-ES <i>Dynamic Niching with Adaptive Niche Radius</i> .	38
Algoritmo 6	–	<i>Nearest-better Clustering (NBC)</i>	39
Algoritmo 7	–	<i>Directed Target to Best Perturbation (DTBP)</i>	41
Algoritmo 8	–	<i>DE/rand/1/bin</i>	69
Algoritmo 9	–	Arcabouço geral do algoritmo proposto	69

SUMÁRIO

1	INTRODUÇÃO	18
1.1	ORGANIZAÇÃO DO TRABALHO	21
2	MODIFICAÇÕES E ADAPTAÇÕES PARA O CMA-ES	22
2.1	<i>COVARIANCE MATRIX ADAPTATION EVOLUTION STRATEGY (CMA-ES)</i>	23
2.2	CMA-ES PARA AMBIENTES MULTIMODAIS	31
2.3	DISCUSSÃO DOS MÉTODOS APRESENTADOS	43
3	<i>LOCAL CMA-ES GUIDED BY DIFFERENTIAL EVOLUTION (L-CMA-DE)</i>	45
3.1	ANÁLISE DA EVOLUÇÃO DO CMA-ES PARA PROBLEMAS MONO-OBJETIVO	45
3.2	COMPORTAMENTO DA MATRIZ DE COVARIÂNCIA	47
3.3	INSERÇÃO DE DIVERSIDADE	50
3.4	CMA-ES COM DIVISÃO DO ESPAÇO DE BUSCA	59
3.5	MODELO PROPOSTO	67
4	VALIDAÇÃO DO MODELO	71
4.1	COMPLEXIDADE DE TEMPO	71
4.2	SENSIBILIDADE PARAMÉTRICA	71
4.2.1	Percentual do Total de Avaliações de Função Remanescentes	73
4.2.2	Quantidade de Gerações do CMA-ES sem Melhorar o Melhor Indivíduo	73
4.2.3	Quantidade de Gerações do DE sem Melhorar o Melhor Indivíduo	74
4.2.4	Grau de Esparsidade da Matriz de Covariância	75
4.2.5	Valor de Referência Considerado como Esparso	76
4.2.6	Quantidade Máxima de Gerações para o DE	76
4.3	AJUSTE PARAMÉTRICO	77
4.4	FUNÇÕES PARA TESTES	80
4.5	ALGORITMOS PARA COMPARAÇÃO	84
4.6	MÉTRICAS	85
4.7	CONFIGURAÇÃO EXPERIMENTAL	86
4.8	RESULTADOS E ANÁLISES	86
4.8.1	Testes com as funções da CEC-2013	87
4.8.2	Testes com as funções da CEC-2017	88

5	CONCLUSÃO E TRABALHOS FUTUROS	108
	REFERÊNCIAS	111

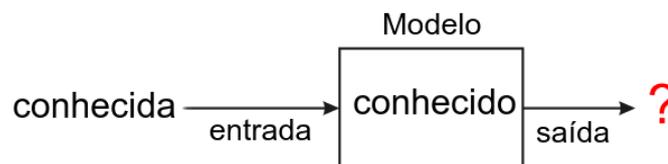
1 INTRODUÇÃO

Algoritmos evolucionários são largamente empregados para resolver diferentes classes de problemas. Eiben e Smith (2015) classifica estes problemas conforme sua natureza, podendo ser de:

- Simulação;
- Modelagem;
- Otimização.

Nos problemas de simulação são caracterizados pela existência de entrada e modelo conhecidos, ao passo que se desconhece a saída, conforme pode ser visto na representação da Figura 1.

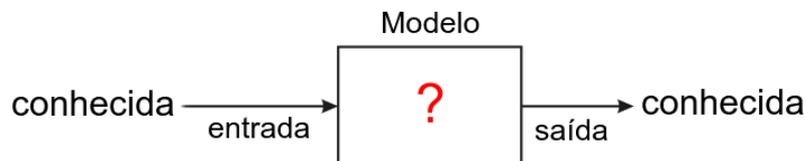
Figura 1 – Arcabouço de um problema de simulação



Fonte: Adaptado de Eiben e Smith (2015).

Já nos problemas de modelagem, tanto a entrada quanto a saída do sistema são conhecidas. O que se espera é conseguir um modelo que possibilite a obtenção da saída desejada, a partir de uma entrada (Figura 2).

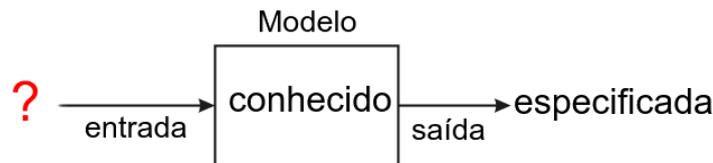
Figura 2 – Arcabouço de um problema de modelagem



Fonte: Adaptado de Eiben e Smith (2015).

Por fim, nos problemas de otimização a entrada é desconhecida, o modelo é conhecido e a saída é especificada, ou seja, se impõe uma saída desejada (ver Figura 3). É neste contexto de problemas que esta dissertação foi desenvolvida. Em outras palavras, deseja-se encontrar as variáveis de decisão que levem à soluções ótimas.

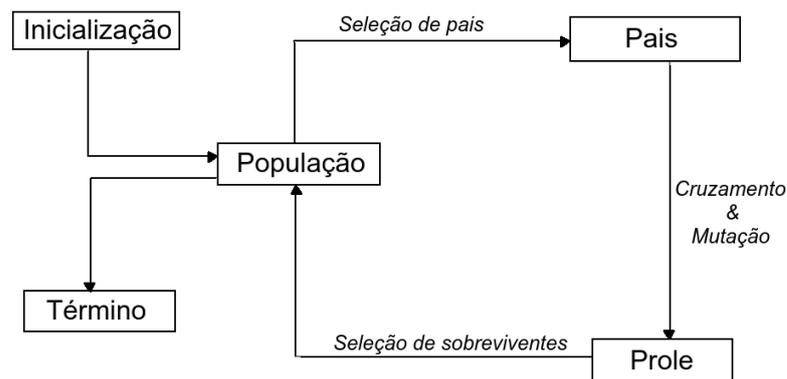
Figura 3 – Arcabouço de um problema de otimização



Fonte: Adaptado de Eiben e Smith (2015).

A Computação Evolucionária é amplamente utilizada para solucionar problemas de otimização. Esta forma de computação tem uma inspiração biológica na teoria da evolução, ou seja, o arcabouço geral consiste em selecionar dois ou mais indivíduos que se cruzam e produzem a prole, a qual passa por um processo de mutação, sendo em seguida selecionados os sobreviventes que darão início a uma nova geração. Este ciclo se repete até que uma condição de parada seja atingida. A Figura 4 ilustra o esquema básico de um algoritmo evolucionário.

Figura 4 – Arcabouço geral de um algoritmo evolucionário



Fonte: Adaptado de Eiben e Smith (2015).

Os problemas de otimização podem ser: mono-objetivo (com uma única função objetivo a ser otimizada); multi-objetivo (MOP), que usualmente possui até 3 funções objetivo; ou de muitos objetivos (MaOP), possuindo acima de 3 funções objetivo. E para lidar com cada uma destas classes de problemas, existem, na Computação Evolucionária, algoritmos específicos: algoritmos evolucionários para um único objetivo; Algoritmos Evolucionários para Objetivos Múltiplos (MOEAs); e Algoritmos Evolucionários para Muitos Objetivos (MaOEAs). Os algoritmos evolucionários ainda podem ser classificados quanto às características dos modelos, sendo os principais: Algoritmos Genéticos, Estratégias de Evolução, Programação Evolucionária, Programação Genética, Sistemas de Classificadores de Aprendizagem, Evolução Diferencial, Otimização de Enxame de Partículas e Algoritmos de Estimção de Distribuição (EIBEN; SMITH, 2015). A classe de interesse para o presente trabalho é a Estratégia de Evolução para problemas de otimização mono-objetivo, onde a representação genética é realizada por números reais.

Para problemas de otimização de um único objetivo no domínio dos números reais, um dos algoritmos amplamente explorado é o *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) (HANSEN; OSTERMEIER, 1996; HANSEN, 2016), que utiliza o conhecimento sobre o deslocamento da população da geração $g - 1$ para a geração g a fim de adaptar a matriz de covariância, a qual será utilizada para gerar os indivíduos da geração $g + 1$, possibilitando acelerar a convergência em direção ao ótimo.

Contudo, existem algumas limitações inerentes ao processo por ele adotado para que se alcance bons resultados em funções mais complexas, como a necessidade de uma população maior e/ou reinicialização do processo evolucionário (HANSEN; KERN, 2004). Procurando entender melhor o processo de evolução simulada com base na matriz de covariância e suas limitações, foi realizada uma revisão da literatura para identificar os pontos de melhorias sugeridas por modelos que se baseiam no CMA-ES. Em seguida, buscando evidenciar as limitações existentes neste algoritmo, bem como identificar em qual parte do processo será realizada a intervenção para melhoria, foram desenvolvidos modelos iniciais que tiveram seus desempenhos comparados ao CMA-ES (HANSEN, 2016).

A literatura apresenta avanços realizados com a utilização de mecanismos que conseguiram mitigar as limitações do CMA-ES, como a reinicialização do processo evolucionário com o aumento da população (AUGER; HANSEN, 2005) e modelos que evoluem a população em nichos, como pode ser visto em Shir e Bäck (2005a).

As análises corroboraram para o conhecimento sobre a forte dependência do CMA-ES ao ponto de inicialização, quando há uma baixa quantidade de indivíduos e a função objetivo é multimodal de alta dimensionalidade, bem como a necessidade de inserção de diversidade de forma a evitar a convergência prematura. Com isso, o objetivo do presente trabalho é construir um modelo baseado no CMA-ES, que harmonize exploração e exploração sem prejuízo do número de avaliações de função, para otimizar funções multimodais de um único objetivo.

Para atingir o objetivo pretendido, desenvolveu-se um algoritmo memético, denominado *Local CMA-ES Guided by Differential Evolution* (L-CMA-DE), que inicializa n CMA-ES independentes com critérios de parada adaptativos, compartilhando informações genéticas com a Evolução Diferencial, do inglês *Differential Evolution* (DE) (EIBEN; SMITH, 2015), a qual é responsável por realizar uma exploração global, evitando que o algoritmo fique preso em um ótimo local. A escolha do DE como mecanismo para gerar diversidade se deu por sua característica exploratória, proveniente dos operadores de variação, conforme discutido no Capítulo 3. Com isso, espera-se que o modelo proposto consiga incorporar exploração mais agressiva, proporcionada pelo mecanismo de mutação do DE, sem reinicialização do processo evolucionário, para acelerar a convergência e melhorar o resultado do processo de otimização.

Foram realizados experimentos comparando o algoritmo proposto com aqueles utilizados como base para sua construção, além de comparações realizadas com outros

modelos inspirados no CMA-ES. Os resultados sugerem que a nova proposta apresenta melhores soluções, quando comparada ao CMA-ES e DE, aplicados isoladamente. Também foi possível observar uma melhor performance do algoritmo proposto, quando comparado à algoritmos Estado-da-Arte, em funções de maior complexidade. Já em funções de menor complexidade, a nova proposta apresenta resultados qualitativos inferiores, quando comparada à algoritmos Estado-da-Arte que também são baseados no CMA-ES.

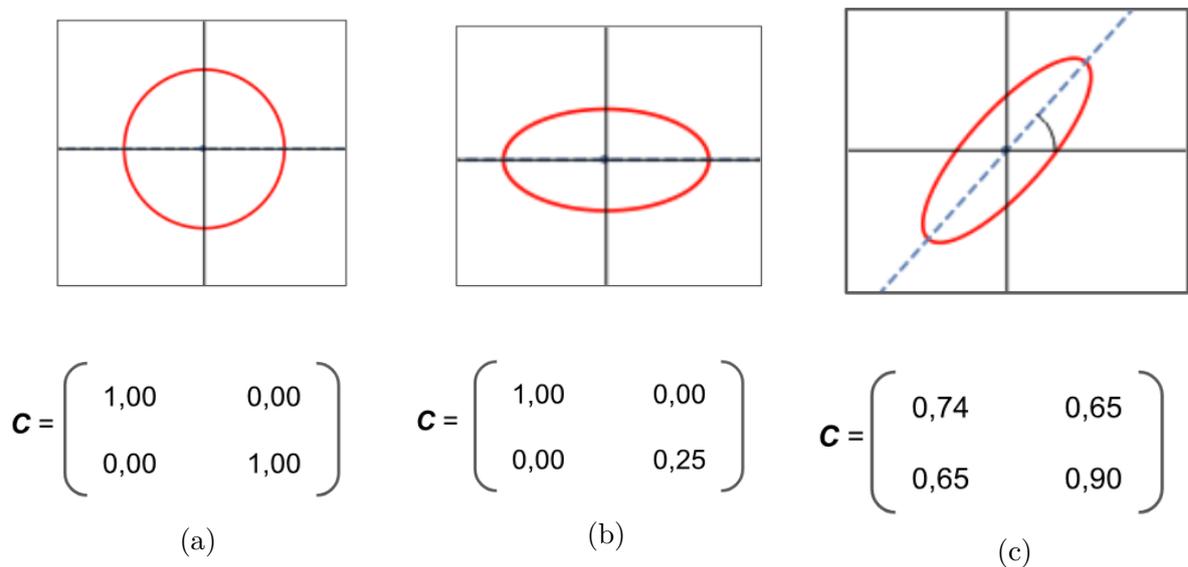
1.1 ORGANIZAÇÃO DO TRABALHO

A presente Dissertação está organizada como se segue: no Capítulo 2 está a revisão de literatura do CMA-ES e alguns outros algoritmos provenientes deste; no Capítulo 3 inicialmente são apresentadas propostas preliminares que subsidiam a construção do L-CMA-DE, descrito ao final do Capítulo; no Capítulo 4 é realizada a validação do modelo proposto, a partir de análise de sensibilidade, ajuste paramétricos e realização de testes; por fim, no Capítulo 5 podem ser vistas as conclusões obtidas a partir do estudo realizado, bem como a projeção para realização de trabalhos futuros.

2 MODIFICAÇÕES E ADAPTAÇÕES PARA O CMA-ES

Conforme Jastrebski e Arnold (2006), Estratégia de Evolução com Adaptação de Matriz de Covariância (*Covariance Matrix Adaptation Evolution Strategy*), CMA-ES (HANSEN; OSTERMEIER, 1996; HANSEN, 2016), é um tipo de Estratégia de Evolução (EE) que reúne informações sobre etapas de pesquisa bem-sucedidas e as utiliza para modificar a matriz de covariância da distribuição das mutações de uma população, de forma direcionada. As mudanças na matriz de covariância são realizadas de tal forma que as variações nas direções do espaço de busca que foram previamente bem sucedidas sejam potencializadas. Para isso são utilizadas informações acumuladas por várias etapas da busca, o que permite adaptar de forma confiável a matriz de covariância, mesmo quando a população é pequena. A Figura 5 ilustra a região de maior probabilidade de gerar soluções (delimitada pela linha vermelha), a partir de uma determinada matriz de covariância (\mathbf{C}), onde é possível observar que os valores de entrada da matriz influencia na distribuição dos dados. Sendo assim, é intuitivo afirmar que seria relevante adaptá-la de forma a maximizar a probabilidade de serem gerados indivíduos nas direções promissoras. E este é o objetivo do CMA-ES.

Figura 5 – Efeito da adaptação da matriz de covariância



Fonte: Adaptado de Pimikov (2018).

Neste Capítulo serão apresentados alguns trabalhos que demonstram a aplicabilidade do CMA-ES, bem como algumas de suas extensões. A Seção 2.1 apresenta em detalhes o funcionamento do CMA-ES original, sendo ao final pontuadas suas eficiências e limitações. Dentre as limitações existentes, o esforço será direcionado de forma a possibilitar um melhor desempenho do CMA-ES em funções multimodais. Com o objetivo

de subsidiar o conhecimento sobre as modificações que podem ser aplicadas ao CMA-ES para que este apresente melhor desempenho neste tipo de função, foi realizado um estudo sobre as estratégias já existentes na literatura que possuem esta finalidade (Seção 2.2). Por fim, a Seção 2.3 apresenta uma breve análise sobre as abordagens descritas.

2.1 COVARIANCE MATRIX ADAPTATION EVOLUTION STRATEGY (CMA-ES)

O CMA-ES é um tipo de EE amplamente utilizada em problemas de otimização mono-objetivo por alcançar resultados factíveis a baixo custo, como pode ser visto em Hansen (2009), Loshchilov et al. (2014), dentre outros.

A ideia principal é que o melhor indivíduo de uma geração g com λ indivíduos, denotado por $\mathbf{x}_{1:\lambda}^{(g)}$, seja selecionado para a geração $g+1$ e dê origem a λ^{g+1} novos indivíduos a partir de perturbações aplicadas a ele. Para o CMA-ES, o indivíduo que dará origem a nova geração será obtido através da média entre os μ melhores da geração atual.

De acordo com Hansen (2016), o CMA-ES gera o k -ésimo indivíduo da próxima geração, denotado por $\mathbf{x}_k^{(g+1)}$, conforme modelo probabilístico definido em (2.1).

$$\mathbf{x}_k^{(g+1)} \sim \mathbf{m}^{(g)} + \sigma^{(g)} \mathcal{N}_k(\mathbf{0}, \mathbf{C}^{(g)}) \quad \text{para } k = 1, \dots, \lambda \quad , \quad (2.1)$$

onde $\mathbf{m}^{(g)}$ representa o vetor médio dos μ indivíduos mais aptos da geração g , $\sigma^{(g)}$ é o passo de mutação para a geração g ; e $\mathcal{N}(\mathbf{0}, \mathbf{C}^{(g)})$ representa a distribuição normal multivariada com média zero e matriz de covariância $\mathbf{C}^{(g)}$.

O indivíduo a ser selecionado para dar origem à próxima geração poderia ser aquele que possuísse a maior aptidão, contudo esta abordagem fica propensa a que frequentemente o algoritmo realize uma convergência prematura, ficando preso em um ótimo local. Uma outra forma de seleção é considerar como melhor indivíduo a média ponderada entre os μ melhores daquela geração, como pode ser visto na Equação (2.1). Esta alternativa não elimina a possibilidade de ficar preso em mínimos (ou máximos) locais, mas aumenta a chance de exploração do ambiente, que pode reduzir a influência de ótimos locais. Sendo assim, Hansen (2016) calcula o vetor médio da geração g de acordo com a Equação (2.2).

$$\mathbf{m}^{(g)} = \sum_{i=1}^{\mu} \omega_i \mathbf{x}_{i:\lambda}^{(g)} \quad , \quad (2.2)$$

sendo:

$$\sum_{i=1}^{\mu} \omega_i = 1, \quad \omega_1 \geq \omega_2 \geq \dots \geq \omega_{\mu} > 0 \quad , \quad (2.3)$$

onde $\mathbf{x}_{i:\lambda}^{(g)}$ o i -ésimo melhor indivíduo da geração g , com respeito à suas respectivas aptidões. Para um problema de minimização tem-se que $f(\mathbf{x}_{1:\lambda}^{(g)}) \leq f(\mathbf{x}_{2:\lambda}^{(g)}) \leq \dots \leq f(\mathbf{x}_{\lambda:\lambda}^{(g)})$, sendo $f(\cdot)$ a função a ser minimizada. Os valores de ω_i obtidos conforme Equação (2.4).

$$\omega_i = \frac{\log(\mu + 0,5) - \log(i)}{\sum_{i=1}^{\mu} \omega_i} \quad (2.4)$$

A combinação dos $\mu^{(g)}$ indivíduos para encontrar-se $\mathbf{m}^{(g)}$ consiste na operação de recombinação. Enquanto que ao assumir $\mathbf{m}^{(g)}$ como o melhor indivíduo, onde a partir do qual serão gerados outros elementos em $g + 1$ (Equação 2.1), define a seleção de sobreviventes do CMA-ES.

Considerando o processo de mutação, representado através da Equação (2.1), percebe-se que a geração de indivíduos está diretamente ligada à matriz de covariância (\mathbf{C}^g). Portanto, adaptá-la pode resultar na geração de uma prole mais apta, sendo este o objetivo do CMA-ES. A matriz de covariância pode ser interpretada como um hiper-elipsoide que representa a densidade de distribuição da população, logo, sua adaptação busca que essa variabilidade seja maximizada nos eixos que geraram melhores indivíduos, aumentando consequentemente as chances de serem gerados indivíduos mais aptos nas próximas gerações.

Hansen (2016) define o cálculo da matriz de covariância de forma empírica (Equação 2.5).

$$\mathbf{C}_{emp}^{(g+1)} = \frac{1}{\lambda - 1} \sum_{i=1}^{\lambda} \left(\mathbf{x}_i^{(g+1)} - \frac{1}{\lambda} \sum_{j=1}^{\lambda} \mathbf{x}_j^{(g+1)} \right) \left(\mathbf{x}_i^{(g+1)} - \frac{1}{\lambda} \sum_{j=1}^{\lambda} \mathbf{x}_j^{(g+1)} \right)^T \quad (2.5)$$

Uma segunda forma, utilizando o vetor médio da geração anterior (Equação 2.2), pode ser aplicada (Equação 2.6).

$$\mathbf{C}_{\lambda}^{(g+1)} = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \left(\mathbf{x}_i^{(g+1)} - \mathbf{m}^{(g)} \right) \left(\mathbf{x}_i^{(g+1)} - \mathbf{m}^{(g)} \right)^T \quad (2.6)$$

As Equações Equação (2.5) e Equação (2.6) podem ser interpretadas de formas diferentes: enquanto \mathbf{C}_{emp} estima a variância dos λ indivíduos gerados, \mathbf{C}_{λ} estima a variância entre as gerações. Como esta última assume que os indivíduos possuem pesos iguais, uma outra abordagem é apresentada em Hansen (2016), na qual são utilizados apenas os μ melhores vetores da geração com suas respectivas ponderações (Equação 2.7).

$$\mathbf{C}_{\mu}^{(g+1)} = \sum_{i=1}^{\mu} \omega_i \left(\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)} \right) \left(\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)} \right)^T \quad (2.7)$$

É possível notar que se a média tomada como referência for da geração g , a variância tende a ser maior que se fosse assumido o vetor médio da geração $g + 1$, ou seja, há uma probabilidade maior de serem gerados indivíduos com maior diversidade.

As atualizações propostas nas Equações (2.5)-(2.7) tipicamente apresentam bom desempenho com populações grandes, impactando diretamente no custo do algoritmo. Com o objetivo de serem utilizadas pequenas populações, algumas técnicas de atualização da matriz de covariância são propostas em Hansen (2016), através da utilização da matriz obtida nas gerações anteriores. Uma destas abordagens pode ser vista na Equação (2.8), onde $\mathbf{C}^{(g+1)}$ é obtida através da média das razões entre a variância estimada dos passos dados pelos μ melhores indivíduos de uma geração $g + 1$, em relação aos μ melhores da geração g , e a variância da geração g .

$$\mathbf{C}^{(g+1)} = \frac{1}{g+1} \sum_{i=0}^g \frac{1}{\sigma^{(i)^2}} \mathbf{C}_\mu^{(i+1)} \quad (2.8)$$

Como ocorre em Equação (2.5) e Equação (2.6), a atualização da matriz de covariância através de Equação (2.8) atribui o mesmo peso para todas as gerações. Para que se ponderar mais fortemente as gerações mais recentes, foi utilizada uma suavização exponencial (Equação 2.9).

$$\begin{aligned} \mathbf{C}^{(g+1)} &= (1 - c_\mu) \mathbf{C}^{(g)} + c_\mu \frac{1}{\sigma^{(g)^2}} \mathbf{C}_\mu^{(g+1)} \\ &= (1 - c_\mu) \mathbf{C}^{(g)} + c_\mu \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda}^{(g+1)} \mathbf{y}_{i:\lambda}^{(g+1)T} \quad , \end{aligned} \quad (2.9)$$

onde $c_\mu \leq 1$ é a taxa de aprendizagem para atualização da matriz de covariância e $\mathbf{y}_{i:\lambda}^{(g+1)} = (\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)}) / \sigma^{(g)}$.

Desta forma, em Hansen (2016), a Equação (2.9) é chamada de *rank- μ -update*. Se a informação for obtida apenas através do melhor indivíduo gerado, chega-se ao *rank-one-update* (Equação 2.10).

$$\mathbf{C}^{(g+1)} = (1 - c_1) \mathbf{C}^{(g)} + c_1 \mathbf{y}_{g+1} \mathbf{y}_{g+1}^T \quad , \quad (2.10)$$

onde c_1 tem a mesma função que c_μ , sendo definido em Hansen (2016) conforme Equação (2.11); $\mathbf{y}_{g+1} = (\mathbf{x}_{1:\lambda}^{(g+1)} - \mathbf{m}^{(g)}) / \sigma^{(g)}$, ou seja, indica o sentido de "sucesso" em relação ao melhor indivíduo da geração $g + 1$. Com isso a matriz de covariância (hiper-elipsoide) se alonga nesta direção, maximizando a probabilidade de serem geradas melhores proles.

$$c_1 = \frac{2}{(N + 1, 3)^2 + \mu_{eff}} \quad , \quad (2.11)$$

sendo N a dimensão do espaço de busca e μ_{eff} definido conforme Equação (2.12), que pode ser interpretado como a massa variável de seleção efetiva.

$$\mu_{eff} = \left(\sum_{i=1}^{\mu} \omega_i^2 \right)^{-1} \quad (2.12)$$

Nas atualizações propostas nas Equações (2.9) e (2.10), a positividade ou negatividade da k -ésima dimensão é desprezada para o cálculo de $\mathbf{C}^{(g+1)}$, uma vez que $\mathbf{y}_{i:\lambda}^{(g+1)} \mathbf{y}_{i:\lambda}^{(g+1)T} = -\mathbf{y}_{i:\lambda}^{(g+1)} - \mathbf{y}_{i:\lambda}^{(g+1)T}$. Ou seja, a matriz de covariância é atualizada através dos valores de entrada da última existente mais a informação sobre as mudanças observadas entre os μ melhores indivíduos de cada geração. Visando um melhor ajuste da matriz de covariância, a informação sobre o caminho de evolução (\mathbf{p}_c) é considerada. Sendo assim, a adaptação da matriz de covariância inclui a tendência desta ao longo das gerações. Assumindo $\mathbf{p}_c^{(0)} = \mathbf{0}$, a atualização do caminho de evolução segue conforme Equação (2.13).

$$\mathbf{p}_c^{(g+1)} = (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2 - c_c)\mu_{eff}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \quad , \quad (2.13)$$

em que $c_c \in [0...1]$ pondera as influências do caminho de evolução e a variação entre os pontos médios das duas últimas gerações; $\sqrt{c_c(2 - c_c)\mu_{eff}}$ é uma constante de normalização, de forma que: caso c_c seja igual a 0, \mathbf{p}_c não sofre alteração. Caso contrário, a influência do deslocamento de $\mathbf{m}^{(g)}$ para $\mathbf{m}^{(g+1)}$ cresce de forma aproximadamente logarítmica com o aumento do valor de c_c , onde sendo este último igual a 1, não há influência de $\mathbf{p}_c^{(g)}$ em $\mathbf{p}_c^{(g+1)}$. Com isso, o *rank-one-update* (Equação 2.10) com a informação sobre o caminho de evolução pode ser reescrito conforme Equação (2.14).

$$\mathbf{C}^{(g+1)} = (1 - c_1)\mathbf{C}^{(g)} + c_1\mathbf{p}_c^{(g+1)}\mathbf{p}_c^{(g+1)T} \quad (2.14)$$

Por fim, a atualização da matriz de covariância provida pelo CMA-ES é realizada através da combinação entre Equação (2.9) e Equação (2.14), considerando tanto informações das μ melhores soluções de uma geração como também a tendência seguida ao longo das gerações (Equação 2.15). Sendo assim, esta será a adaptação utilizada no algoritmo proposto nesta dissertação.

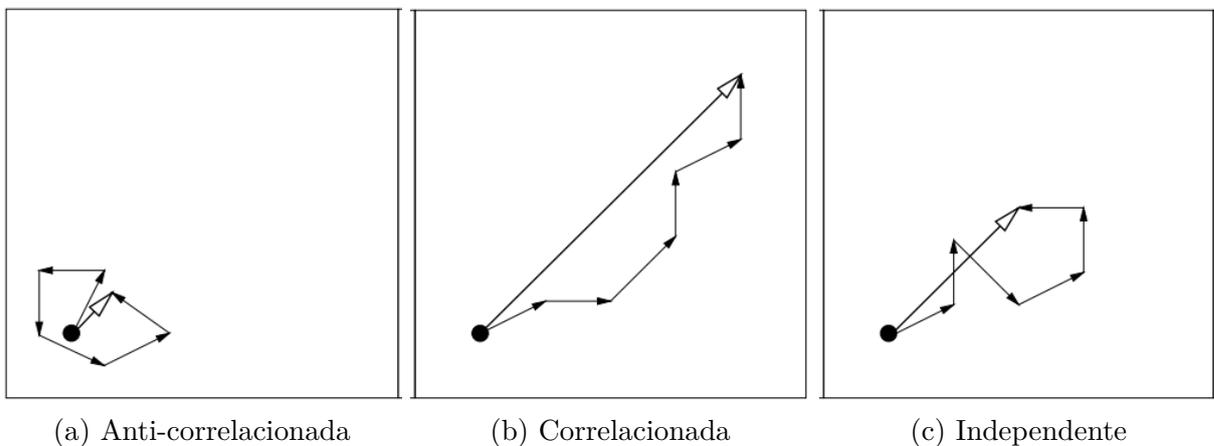
$$\begin{aligned} \mathbf{C}^{(g+1)} = & (1 - c_1 - c_\mu)\mathbf{C}^{(g)} \\ & + c_1\mathbf{p}_c^{(g+1)}\mathbf{p}_c^{(g+1)T} + c_\mu \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda}^{(g+1)} \left(\mathbf{y}_{i:\lambda}^{(g+1)}\right)^T \end{aligned} \quad (2.15)$$

Além da adaptação da matriz de covariância apresentada, o CMA-ES propõe uma adaptação do passo de mutação (σ) definida independentemente da adaptação de \mathbf{C} . A adaptação do passo de mutação visa aumentar as chances de gerarem-se indivíduos nas direções promissoras. A abordagem para atualizar o passo de mutação de uma geração busca evitar o acréscimo de novos parâmetros no processo evolucionário ou a necessidade de grandes populações para sua definição.

Sendo assim, Hansen (2016) propôs um tipo de controle do passo de mutação chamado de não-aleatório, em que os passos são atualizados através de uma relação entre gerações sucessivas, utilizando o caminho de evolução (semelhante ao caminho de evolução

da matriz de covariância, apresentado anteriormente) permitindo seu uso em populações grandes ou pequenas. O método desenvolvido é denominado de *cumulative path length control* (controle cumulativo do comprimento do caminho), *cumulative step-size control* (controle cumulativo do passo de mutação) ou *Cumulative Step-size Adaptation* (CSA) (adaptação cumulativa do passo de mutação). A Figura 6 ilustra as possibilidades de configuração da trajetória de evolução da população, onde os vetores com a seta preenchida representam a trajetória entre duas gerações consecutivas, e os vetores com a seta vazada representam os vetores resultantes da soma das trajetórias realizadas.

Figura 6 – Configurações de trajetória de evolução



Fonte: Hansen (2016).

As informações sobre o tamanho do caminho de evolução para o passo de mutação é utilizada sob as seguintes perspectivas:

- Se o caminho de evolução for curto (Figura 6a), espera-se que o fator provocador seja o fato dos deslocamentos estarem cancelando-se mutuamente (anti-correlacionados), logo o passo de mutação deve ser diminuído;
- Se o caminho de evolução for longo (Figura 6b), provavelmente os passos devem estar seguindo para a “mesma direção” (correlacionados), então o passo será aumentado com a expectativa de celeridade do processo;
- E no melhor caso, os passos são (aproximadamente) perpendiculares (descorrelacionados) (Figura 6c).

Sendo assim, Hansen (2016) propõe a utilização do caminho de evolução para o passo de mutação (\mathbf{p}_σ), que é atualizado de forma semelhante à atualização do caminho de evolução para a matriz de covariância (Equação 2.13), a fim de melhor adaptá-lo. Contudo, diferente da atualização de \mathbf{p}_c , agora busca-se escolher o tamanho de \mathbf{p}_σ independente de

direção, a fim de mitigar o problema relacionado à correlação das trajetórias. Com isso, o caminho de evolução para o passo de mutação é dado conforme Equação (2.16).

$$\mathbf{p}_\sigma^{(g+1)} = (1 - c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)\mu_{eff}}\mathbf{C}^{(g)-\frac{1}{2}}\frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}, \quad (2.16)$$

onde:

$$\begin{aligned} \mathbf{C}^{(g)-\frac{1}{2}} &= \mathbf{B}^{(g)} \times \mathbf{D}^{(g)-1} \times \mathbf{B}^{(g)T} \\ &= \mathbf{B}^{(g)} \text{diag} \left(\frac{1}{d_1^{(g)}}, \dots, \frac{1}{d_n^{(g)}} \right) \mathbf{B}^{(g)T}, \end{aligned} \quad (2.17)$$

sendo $\mathbf{B}^{(g)}$ a matriz com autovetores de $\mathbf{C}^{(g)}$, dispostos em colunas; $\mathbf{D}^{(g)}$ a matriz diagonal com a raiz quadrada dos autovalores associados aos autovetores em $\mathbf{B}^{(g)}$. Por fim, a autodecomposição de $\mathbf{C}^{(g)}$ pode ser realizada através da Equação (2.18)

$$\mathbf{C}^{(g)} = \mathbf{B}^{(g)} \times \mathbf{D}^{(g)2} \times \mathbf{B}^{(g)T} \quad (2.18)$$

O motivo do operador $\mathbf{C}^{(g)-\frac{1}{2}}$ se dá para que se produza o tamanho esperado de $\mathbf{p}_\sigma^{(g+1)}$ independente de sua direção, onde a Equação (2.17) pode ser vista da seguinte forma:

- $\mathbf{B}^{(g)T}$ rotaciona o espaço formado pelas colunas de $\mathbf{B}^{(g)}$;
- $\mathbf{D}^{(g)-1}$ aplica uma transformação de forma que os eixos se tornam igualmente dimensionados;
- $\mathbf{B}^{(g)}$ rotaciona o resultado para o sistema de coordenadas original.

Para atualizar $\sigma^{(g)}$, verifica-se a diferença entre os tamanhos de $\mathbf{p}_\sigma^{(g+1)}$ e o esperado, de forma que $\sigma^{(g+1)}$ pode ser obtido através de Equação (2.19).

$$\sigma^{(g+1)} = \sigma^{(g)} \times \exp \left[\frac{c_\sigma}{d_\sigma E \|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} \left(\|\mathbf{p}_\sigma^{(g+1)}\| - E \|\mathcal{N}(\mathbf{0}, \mathbf{I})\| \right) \right], \quad (2.19)$$

onde $d_\sigma \approx 1$, é um parâmetro de suavização e $E \|\mathcal{N}(\mathbf{0}, \mathbf{I})\|$ é a esperança da norma Euclidiana de um vetor $\mathcal{N}(\mathbf{0}, \mathbf{I})$ aleatoriamente distribuído. Após algumas operações algébricas, a Equação (2.19) pode ser reescrita com Equação (2.20).

$$\sigma^{(g+1)} = \sigma^{(g)} \times \exp \left[\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma^{(g+1)}\|}{E \|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1 \right) \right] \quad (2.20)$$

Desta forma é possível perceber que se a norma de $\mathbf{p}_\sigma^{(g+1)}$ for maior que a norma do vetor esperado, então $\sigma^{(g)}$ será decrementado. Se o contrário ocorrer, então este deverá ser aumentado. E caso $\|\mathbf{p}_\sigma^{(g+1)}\| = E \|\mathcal{N}(\mathbf{0}, \mathbf{I})\|$, o passo de mutação não é alterado.

Em suma, através do controle de apenas três parâmetros principais ($\mathbf{m}, \mathbf{C}, \sigma$), o CMA-ES consegue abrandar, ainda que de forma não tão eficaz (como será visto no Capítulo 4), algumas das limitações inerentes a este tipo de algoritmo evolucionário:

- Baixa performance em funções objetivo difíceis. Minimizada com a atualização definida pela Equação (2.15);
- Necessita de uma população grande, principalmente para problemas de difícil solução. As taxas de aprendizagem c_1 e c_μ (Equação 2.15) fazem a ponderação entre a importância dos melhores indivíduos mais recentes e a tendência observada;
- Convergência prematura. Através do controle do passo de mutação (Equação 2.20).

Para facilitar o entendimento, o pseudocódigo do CMA-ES pode ser visto no Algoritmo 1, no qual os parâmetros e as variáveis são assim definidos: n é o tamanho da população; $minErr$ é o menor erro aceito como critério de parada do algoritmo; $maxAval$ é o número máximo de avaliações de função para o encerramento do algoritmo; σ é o passo de mutação; λ é o tamanho da população; μ é a quantidade dos melhores indivíduos a serem levados em consideração para o processo de adaptação; $limInf$ e $limSup$ são os limite inferior e superior para as dimensões do problema, respectivamente; \mathbf{x}_{med} é o ponto médio dentre os μ melhores indivíduos de uma determinada geração; d é a dimensão das variáveis; \mathbf{C} é a matriz de covariância; \mathbf{B} é matriz que define as coordenadas do sistema, composta pelos autovetores de \mathbf{C} dispostos em colunas; \mathbf{D} é matriz diagonal composta pelos autovalores de \mathbf{C} associados aos autovetores em \mathbf{B} , definindo a potencialização em cada sentido; \mathbf{I} é uma matriz identidade; \mathbf{p}_c é o caminho de evolução da matriz de covariância; \mathbf{p}_s é o caminho de evolução do passo de mutação; \mathbf{x}_k é o k -ésimo indivíduo de uma determinada geração; $\mathbf{X}_{1,\mu}$ representa o conjunto composto pelos μ melhores indivíduos de uma determinada geração, com respeito à suas aptidões. A Figura 7 ilustra a distribuição da população durante a evolução da população do CMA-ES.

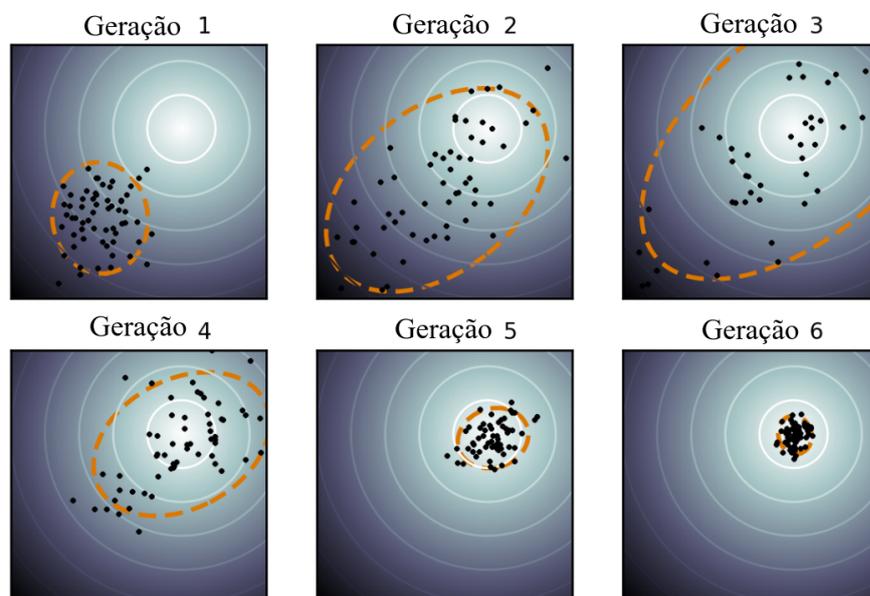
Algoritmo 1 CMA-ES

```

1: inicializarParametros( $n, minErr, maxAval, \sigma, \lambda, \mu, limInf, limSup$ )
2:  $\mathbf{x}_{med} \leftarrow geraVetorUniforme(d, limInf, limSup)$ 
3:  $\mathbf{C} \leftarrow \mathbf{I}$ 
4:  $\mathbf{B} \leftarrow \mathbf{I}$ 
5:  $\mathbf{D} \leftarrow \mathbf{I}$ 
6:  $\mathbf{p}_c \leftarrow geraVetorNulo(d)$ 
7:  $\mathbf{p}_s \leftarrow geraVetorNulo(d)$ 
8:  $contAval \leftarrow 0$  ▷ quantidade de avaliações de função computadas
9: enquanto  $contAval < maxAval$  faça
10:   para  $k \leftarrow 1$  até  $\lambda$  faça
11:      $\mathbf{x}_k \leftarrow gerarProle(\mathbf{x}_{med}, \sigma, \mathbf{B}, \mathbf{D})$ 
12:     Computa a aptidão de  $\mathbf{x}_k$ 
13:      $contAval \leftarrow contAval + 1$ 
14:   fim para
15:    $\mathbf{x}_{med} \leftarrow calculaMedia(\mathbf{X}_{1:\mu})$ 
16:    $\mathbf{p}_s \leftarrow atualizaCaminhoSigma(\mathbf{B})$ 
17:    $\mathbf{p}_c \leftarrow atualizaCaminhoMatrizCovariancia(\mathbf{B}, \mathbf{D})$ 
18:    $\mathbf{C} \leftarrow adaptaMatrizCovariancia(\mathbf{C}, \mathbf{B}, \mathbf{D}, \mathbf{p}_c)$ 
19:    $\sigma \leftarrow adaptaSigma(\sigma, \mathbf{p}_s)$ 
20:    $\mathbf{B} \leftarrow geraAutovetores(\mathbf{C})$ 
21:    $\mathbf{D} \leftarrow geraAutovalores(\mathbf{C})$  ▷ valores na diagonal
22:    $\mathbf{D} \leftarrow \sqrt{\mathbf{D}}$  ▷  $\mathbf{D}$  agora contém os desvios padrão
23:   se  $calculaErr(\mathbf{x}_1) \leq minErr$  então
24:     pare ▷ O algoritmo é encerrado
25:   fim se
26: fim enquanto
27: imprime( $\mathbf{x}_1$ )

```

Figura 7 – Evolução da população do CMA-ES



Fonte: Adaptado de Khan (2018).

Apesar de um bom desempenho já alcançado por diferentes versões do CMA-ES, este algoritmo possui algumas limitações, como a atualização do passo de mutação que perde eficácia quando as funções objetivo são ruidosas (BEYER; ARNOLD, 2003). Uma outra limitação inerente à versão inicial e suas variantes se dá pelo fato do algoritmo ficar preso em mínimos locais com grande frequência, uma vez que o processo evolucionário é ajustado com vistas aos μ melhores indivíduos de uma geração e o caminho percorrido por eles ao longo das gerações. Ou seja, em funções multimodais como a função *Ackley* (POTTER; JONG, 1994) não espera-se um bom desempenho do CMA-ES caso ele possua uma população pequena (HANSEN; KERN, 2004). Portanto, modelos com exploração mais agressiva foram propostos com o intuito de mitigar esta ineficiência, sendo algumas delas tratados na Seção 2.2.

2.2 CMA-ES PARA AMBIENTES MULTIMODAIS

Procurando melhorar o desempenho do CMA-ES original em tratar problemas de otimização cuja função objetivo é multimodal, como é o caso da função *Ackley*, foram desenvolvidos algoritmos com estratégias distintas. Nesta Seção serão apresentados trabalhos que propuseram esta melhoria utilizando uma das seguintes ações: reinicialização, abordagens baseadas em nichos, inserção de imigrantes ou hibridização.

Quanto à métodos de reinicialização, um dos algoritmos desenvolvidos é o *restart- (μ_W, λ) -CMA-ES with Increasing Population* (IPOP-CMA-ES), proposto em Auger e Hansen (2005), que reinicia autonomamente a simulação da evolução, aumentando a população inicial entre 1,5 a 5 vezes da anterior, sempre que algum dos critérios considerados for satisfeito, como:

- A variação entres as melhores aptidões obtidas em um determinado número de gerações é igual a zero, ou a variação entre as melhores aptidões da última geração está abaixo de um limiar;
- Se o desvio padrão da distribuição Normal for menor que um determinado limiar em todas as coordenadas e σp_c for menor que este mesmo limiar, em todas as componentes.
- Se a adição de um vetor com desvio padrão 0,1 na direção do eixo principal de $\mathbf{C}^{(g)}$ não alterar o vetor média obtido até então;
- Se a adição de um desvio padrão 0,2 em cada coordenada não alterar o vetor média obtido;
- Se o número de condição da matriz de covariância \mathbf{C} (FRANCO, 2006), definido conforme Equação (2.21), estiver acima de um limiar.

$$\text{cond}(\mathbf{C}) = \|\mathbf{C}\| \times \|\mathbf{C}^{-1}\| \quad , \quad (2.21)$$

onde $\|\cdot\|$ representa a norma da matriz.

Também são sugeridos alguns critérios para finalização do algoritmo:

- Após $n \times 10^4$ avaliações de função, onde n representa o número de dimensões do problema;
- Se o valor do erro da função objetivo estiver abaixo de 10^{-8} .

A motivação para esta reinicialização aleatória é tornar a exploração mais agressiva, diminuindo a probabilidade de ficar preso a mínimos locais. A reinicialização com aumento da população, apesar de possibilitar maior exploração, afeta diretamente no custo do algoritmo, aumentando a quantidade de avaliações de função. Deve-se também ressaltar o fato que o CMA-ES em sua (re)inicialização gera um ponto médio, através de distribuição uniforme dentro de limites estabelecidos, e só a partir de então são gerados os indivíduos, ou seja, se nas reinicializações do IPOPOP-CMA-ES este ponto for gerado em lugar próximo a mínimos locais, o algoritmo tende a ficar preso neles. Como alternativa foram desenvolvidos alguns algoritmos, como: *BI-population Covariance Matrix Adaptation Evolution Strategy* (BIPOP-CMA-ES) (HANSEN, 2009) e *New BI-population weighted-aCMA-ES with Increasing Population* (NBIPOP-aCMA-ES) (LOSHCHILOV, 2013). Agora ao invés de reinicializar todo o processo de evolução, a população é incrementada ou decrementada a cada geração até que um determinado critério de parada seja atingido, para só a partir de então reinicializar-se todo o processo evolucionário com a “melhor” técnica.

Hansen (2009) utiliza o CMA-ES, inicializando a população padrão $\lambda_{def} = 4 + \lfloor 3 \times \ln(D) \rfloor$, onde D é a dimensão de cada indivíduo. Assim, o algoritmo é reinicializado sempre que um critério de parada é atingido. Os autores utilizaram dois métodos de reinicialização (aumentando e reduzindo o tamanho da população), que são aplicados de forma intercalada sendo armazenadas as avaliações de função acumuladas em cada um deles (custo). Ao final das 9 reinicializações, o algoritmo passa a aplicar apenas o método que obteve menor quantidade acumulada de avaliações de função. Para o método de aumento da população, esta é dobrada e para a redução (λ_s) segue conforme Equação (2.22).

$$\lambda_s = \lfloor \lambda_{def} \times \left(\frac{1}{2} \times \frac{\lambda_l}{\lambda_{def}} \right)^{u[0,1]^2} \rfloor \quad , \quad (2.22)$$

sendo l o tamanho da última população, e $u[0, 1]$ denota número aleatório gerado através de distribuição Uniforme com limites 0 e 1.

Durante os testes realizados observou-se um desempenho satisfatório em funções multimodais irregulares. Apresentando, entretanto, um baixo desempenho quando as funções representam um ambiente regular ou com suavização.

Já a estratégia de reinicialização NBIPOP-aCMA-ES ajusta de forma adaptativa a quantidade máxima de avaliações de função a ser permitida para cada um dos métodos de reinicialização, onde aquele que obtiver a melhor solução encontrada até o momento passa a ter sua quantidade máxima de avaliações de função incrementada, diferentemente do BIPOP-CMA-ES onde cada método de reinicialização possui a mesma quantidade máxima de avaliações de funções. Além disso, também é realizada a adaptação do passo de mutação, o qual é decrementado a partir de um fator ($k_{\sigma_{dec}}$). Os autores sugerem que o valor do passo de mutação inicial (σ^0) seja um valor pequeno, sob o argumento de que o CMA-ES com um passo de mutação inicial relativamente pequeno é capaz de explorar pequenas bacias de atração, enquanto que com o valor inicial elevado negligencia a estrutura local da função. Além desta justificativa, os autores concluem a fundamentação para iniciar com um valor pequeno afirmando que, inicialmente, um tamanho de passo relativamente pequeno aumentará rapidamente se a população for se deslocando em sentidos correlacionados (Figura 6b), permitindo que o algoritmo recupere as mesmas propriedades de pesquisa global. Além disso, o NBIPOP-aCMA-ES aplica uma adaptação da matriz de covariância chamada de *Weighted Active Covariance Matrix Adaptation Evolution Strategy* ($(\mu/\mu_w, \lambda)$ -aCMA-ES) (HANSEN; ROS, 2010), a qual maximiza as chances de serem geradas soluções em direções promissoras e reduz a probabilidade de gerar soluções nas direções que apresentem um histórico de indivíduos ruins.

Loshchilov (2013) ainda faz um estudo comparativo entre outras abordagens, sejam elas: *BI-population weighted-aCMA-ES* (BIPOP-aCMA-ES), que não adapta o passo de mutação; o *restart- (μ_w, λ) -weighted-aCMA-ES with Increasing Population* (IPOP-aCMA-ES), derivado do IPOP-CMA-ES, utiliza o $(\mu/\mu_w, \lambda)$ -aCMA-ES para adaptar a matriz de covariância; e o *New weighted-aCMA-ES with Increasing Population* (NIPOP-aCMA-ES), que além de contemplar a realização do IPOP-aCMA-ES, adapta o passo de mutação, decrementando-o por um fator k_{dec} (configurado para 1.6). Contudo, o NBIPOP-aCMA-ES foi o algoritmo que obteve o melhor desempenho (convergência e quantidade de avaliações de função).

Visando melhor adaptar o passo de mutação, quando o algoritmo é reinicializado, Yamaguchi e Akimoto (2017) apresenta um método que aplica ao IPOP-CMA-ES e ao BIPOP-CMA-ES (chamados aqui de estratégias auxiliares) uma nova estratégia de reinicialização que se utiliza de memória para armazenar a distribuição dos dados quando um critério de reinicialização destes algoritmos é atingido. Nesta nova concepção, o algoritmo mantém um histórico com os vetores de média e matrizes de covariâncias encontradas quando o critério de reinicialização das estratégias auxiliares é atingido. A partir de então, a cada geração, calcula-se a Divergência de Kullback-Leibler (KL) (KULLBACK, 1997) entre a distribuição atual e aquelas armazenadas em memória, e se esta for menor ou igual a um limiar, a população é reinicializada e a informação desta similaridade é armazenada em memória. Caso seja alcançada uma quantidade previamente estipulada de semelhan-

ças, assume-se que o σ inicial é muito elevado e está fazendo com que o algoritmo saia de um local promissor, então o próximo passo de mutação sofrerá um decréscimo.

Além de utilizar métodos de reinicialização para evitar que o CMA-ES fique preso em um ótimo local, pode-se provocar diversidade explotando em regiões distintas do espaço de busca através de nichos. A fundamentação deste tipo de ação é factível porque o CMA-ES é conhecido por seu bom desempenho em problemas convexos, contudo não apresenta boa acurácia em ambientes multimodais (PREUSS, 2010), e com a separação da população em nichos espera-se que cada subpopulação explore locais diferentes do ambiente, minimizando o risco de ficar preso em ótimos locais. E esta é a proposta apresentada em Shir e Bäck (2005a), o qual se originou do *ES dynamic niching algorithm* (SHIR; BÄCK, 2005b) que cria uma população através de nichos, evoluindo e atualizando-os a cada geração, conforme Algoritmo 2.

Algoritmo 2 *ES Dynamic Niching Algorithm*

- 1: *mutaPopulacao*
 - 2: *ordenaPopulacao* ▷ pela aptidão
 - 3: **para** $i \leftarrow 1$ **até** q **faça** ▷ q é a quantidade de nichos desejados
 - 4: Gera $\bar{\lambda}_i$ filhos conforme segue:
 - 5: Escolhe o primeiro pai por seleção através de torneio
 - 6: Escolhe o melhor indivíduo do nicho como o segundo pai
 - 7: Aplica recombinação
 - 8: Seleciona os η melhores dentre os $\bar{\lambda}_i$ filhos e os $\mu - \eta$ indivíduos do nicho i para formarem a próxima geração
 - 9: **fim para**
 - 10: Gera randomicamente $\omega = \mu$ indivíduos adicionais
 - 11: Une todos os q nichos, para formar a próxima geração
-

A diferença do algoritmo proposto em Shir e Bäck (2005a) para o seu predecessor (SHIR; BÄCK, 2005b) é o fato de que no primeiro são utilizados os operadores de recombinação intermediária e mutação auto-adaptativa; ao passo de que agora é aplicado o CMA-ES para este fim, mais precisamente $(1, \lambda)$ -CMA, conforme Algoritmo 3.

Algoritmo 3 *Dynamic Niching* com $(1, \lambda)$ -CMA-ES: uma única iteração

```

1: para  $i \leftarrow 1$  até  $q + 1$  faça ▷ pontos
2:   Gera  $\lambda$  proles através do CMA aplicado à  $\mathbf{x}_i$ 
3: fim para
4: Comuta da aptidão da população
5:  $DPI(\lambda, (q + 1), TamanhoPopulacao, q, raioDoNicho)$  ▷ detectar picos
6: para  $j \leftarrow 1$  até  $q$  faça
7:   Assume o pico como sendo o ponto central para a próxima geração
8:   Gera as  $\lambda_j$  proles
9: fim para
10: se  $N_{dps} < q$  então ▷  $N_{dps}$  é a quantidade de picos gerados
11:   Gera  $(q - N_{dps})$  novos picos
12:   Gera as proles a partir de cada um dos  $(q - N_{dps})$  novos picos
13: fim se
14: Desconsidera o  $(q + 1)$ -ésimo pico
15: Gera 1 pico de forma aleatória

```

Para o *Dynamic Niching* com adaptação da matriz de covariância são criados $q + 1$ nichos, onde q nichos são gerados através do método *DPI* (Algoritmo 4), com raio de atuação ρ , mais 1 nicho que não leva em consideração esta abordagem de identificação de picos.

Algoritmo 4 *Dynamic Peak Identification* (DPI)

```

1: entrada:    $Pop$  - conjunto com toda a população
2:            $N$  - tamanho da população
3:            $q$  - número de picos desejados
4:            $\rho$  - raio do nicho
5: Ordena a população em ordem decrescente de aptidão
6:  $i \leftarrow 1$ 
7:  $NumPicos \leftarrow 0$ 
8:  $DPS \leftarrow \emptyset$  ▷ Conjunto Dinâmico de Picos
9: enquanto  $NumPicos < q$  ou  $i < N + 1$  faça
10:  se  $Pop[i]$  não está dentro do raio  $\rho$  de um pico em  $DPS$  então
11:     $DPS \leftarrow DPS \cup Pop[i]$ 
12:     $NumPicos \leftarrow NumPicos + 1$ 
13:  fim se
14:   $i \leftarrow i + 1$ 
15: fim enquanto
16: saída:    Conjunto Dinâmico de Picos

```

O valor de ρ depende do número de picos a serem identificados, sendo obtido através de $\rho = r / \sqrt{q}$; em que dados os menores e maiores valores $x_{k,min}$ e $x_{k,max}$ da k -ésima coordenada do indivíduo \mathbf{x} no espaço de decisão, r pode ser definido conforme Equação (2.23).

$$r = \frac{1}{2} \times \sqrt{\sum_{k=1}^n (x_{k,min} - x_{k,max})^2} \quad (2.23)$$

Com o objetivo de provocar ainda mais diversidade, Shir e Bäck (2006) apresenta uma proposta de modificação ao *ES Dynamic Nicheing Algorithm*, onde um gene com o raio de ação para compartilhamento de aptidão (*Fitness Sharing*) é adicionado ao indivíduo. Sendo este parâmetro atualizado de forma adaptativa. Para tanto, a função de compartilhamento de aptidão se dá conforme Equação (2.24).

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\rho}\right)^{\zeta_{sh}} & \text{se } d_{ij} < \rho \\ 0 & \text{caso contrário} \end{cases}, \quad (2.24)$$

onde d_{ij} é a distância entre os indivíduos i e j , ρ é o raio para todos os indivíduos daquele nicho e ζ_{sh} é um parâmetro de controle, onde sendo 1 a função é linear, mas para valores maiores o efeito de indivíduos semelhantes é menor. Logo, o cômputo para cada indivíduo é dada pela Equação (2.25).

$$m_i = \sum_{j=1}^N sh(d_{ij}) \quad , \quad (2.25)$$

onde N é a quantidade de indivíduos na população.

Com isso, a aptidão compartilhada do i -ésimo indivíduo é obtida conforme Equação (2.26).

$$f_i^{sh} = \frac{f_i}{m_i} \quad , \quad (2.26)$$

onde f_i denota a aptidão do i -ésimo indivíduo.

Considerando a abordagem demonstrada, para obter-se o compartilhamento dinâmico entre nichos, o efeito computado para cada indivíduo (m_i^{din}) é obtido através da Equação (2.27).

$$m_i^{din} = \begin{cases} n_j & \text{se o indivíduo } i \text{ está no nicho } j \\ m_i & \text{caso contrário (indivíduo não-pico)} \end{cases}, \quad (2.27)$$

onde n_j é a quantidade de indivíduos no j -ésimo nicho. O termo “indivíduo não-pico” se refere àqueles indivíduos gerados de forma aleatória, sem que se leve em consideração os picos obtidos através do Algoritmo 4. Com isso, a aptidão compartilhada para o i -ésimo indivíduo é descrita conforme Equação (2.28).

$$f_i^{din} = \frac{f_i}{m_i^{din}} \quad (2.28)$$

Conforme mencionado inicialmente, a proposta de Shir e Bäck (2006) é alterar o modo de calcular/atualizar o raio ρ , fazendo com que esta alteração seja realizada de forma adaptativa, onde cada indivíduo carregaria consigo (em um novo gene) a informação do seu raio (ρ) de compartilhamento de aptidão. Onde o raio do i -ésimo indivíduo na

geração $g + 1$ (ρ^{g+1}) pode ser obtido utilizando o raio de seu predecessor (ρ_{parent}^g), bem como de seu novo passo (σ_{parent}^{g+1}) (Equação 2.29).

$$\rho_i^{g+1} = (1 - c_i^{g+1}) \times \rho_{parent}^g + c_i^{g+1} \times \sigma_{parent}^{g+1} \quad , \quad (2.29)$$

onde c_i^{g+1} é o coeficiente de aprendizagem individual que é atualizado de acordo com a variação do tamanho do passo de mutação (Equação (2.30)).

$$c_i^{g+1} = \frac{1}{5} \times (1 - \exp\{\zeta \times \Delta\sigma_i^{g+1}\}) \quad \Delta\sigma_i^{g+1} = |\sigma_{parent}^{g+1} - \sigma_{parent}^g| \quad (2.30)$$

Como pode ser visto na Equação (2.30), este coeficiente tende à zero, à medida que a diferença entre os passos é pequena, o que sinaliza a convergência do algoritmo. Logo, o parâmetro ζ é aplicado para potencializar esta diferença entre os passos. Após a identificação dos picos e divisão da população em seus respectivos nichos, uma nova função para a redução da aptidão é aplicada (Equação (2.31)).

$$g(m_i^{din}, \lambda_{exp}) = 1 + \Theta(\lambda_{exp} - m_i^{din}) \times \frac{(\lambda_{exp} - m_i^{din})^2}{\lambda_{exp}} + \Theta(m_i^{din} - \lambda_{exp}) \times (\lambda_{exp} - m_i^{din})^2 \quad , \quad (2.31)$$

onde $\Theta(\cdot)$ denota a função degrau e λ_{exp} é a quantidade esperada de indivíduos por nicho. Logo, é possível observar que com $m_i^{din} = \lambda_{exp}$ o resultado da função é 1 e a aptidão do indivíduo não será alterada (Equação 2.32). Caso contrário, este sofrerá uma penalização de $(\lambda_{exp} - m_i^{din})^2$ para aqueles indivíduos pertencentes à nichos que ultrapassem λ_{exp} elementos. Em contrapartida, os indivíduos que compõem nichos cujo tamanho seja menor que λ_{exp} sofreram uma pequena penalização na ordem de $(\lambda_{exp} - m_i^{din})^2 / \lambda_{exp}$.

Baseados nestas propriedades, Shir e Bäck (2006) definem a aptidão compartilhada do indivíduo i de um determinado nicho seguindo a Equação (2.32), sendo a seleção de sobreviventes para cada nicho baseada neste compartilhamento. O algoritmo proposto em Shir e Bäck (2006) pode ser sumarizado no Algoritmo 5.

$$f_i^{nicho} = \frac{f_i}{g(m_i^{din}, \lambda_{exp})} \quad (2.32)$$

Algoritmo 5 $(1, \lambda)$ -CMA-ES *Dynamic Niching with Adaptive Niche Radius*

- 1: **para** $i \leftarrow 1$ **até** $q+p$ **faça** ▷ pontos
 - 2: Gera λ proles através do CMA aplicado à i
 - 3: Atualiza o raio ρ_i^{g+1} ▷ Equação (2.29)
 - 4: **fim para**
 - 5: Computa da aptidão da população
 - 6: $DPI(\lambda, (q+1), TamanhoPopulacao, q, raioDoNicho)$ ▷ detectar picos
 - 7: Calcula a influência da vizinhança ▷ Equação (2.27)
 - 8: **para** $j \leftarrow 1$ **até** $q+p$ **faça**
 - 9: Computa a aptidão do nicho j ▷ Equação (2.32)
 - 10: Assume o indivíduo com o melhor aptidão do j -ésimo nicho como pico para a próxima geração
 - 11: Gera as λ_j proles
 - 12: **fim para**
 - 13: **se** $N_{dps} =$ quantidade de picos $< q$ **então**
 - 14: Gera $q - N_{dps}$ novos picos
 - 15: Gera as proles a partir de cada um dos $q - N_{dps}$ novos picos
 - 16: **fim se**
 - 17: Desconsidera o $(q+1)$ -ésimo \dots $(q+p)$ -ésimo pico
 - 18: Gera $(q+p)$ picos de forma aleatória
-

É possível perceber no Algoritmo 5 que há uma outra diferença, em comparação ao Algoritmo 3, que é a quantidade de nichos aleatórios (p) é ajustável, permitindo uma maior diversidade.

Com o objetivo de melhor adaptar os nichos criados, Preuss (2010) propõe um outro tipo de abordagem, onde são criados indivíduos representantes de cada nicho, sendo-lhes aplicado CMA-ES para gerar os indivíduos pertencentes a suas respectivas subpopulações. Neste momento é aplicada a *Nearest-better Clustering* (Algoritmo 6), que se utiliza da heurística *Spanning Tree* para interligar cada indivíduo ao seu melhor vizinho mais próximo. Após aplicar o método de ligação entre os indivíduos, as arestas que forem maiores que um determinado valor ψ (Equação 2.33) são eliminadas, sendo assim formados os nichos.

$$\psi = \phi \times \frac{1}{n} \sum_{k=1}^n a_k \quad , \quad (2.33)$$

onde ϕ é um parâmetro a ser estimado, que segundo Preuss (2010) o valor 2 é a escolha adequada; a_k é a distância entre o indivíduo k e seu melhor vizinho mais próximo; e n é o tamanho da população atual.

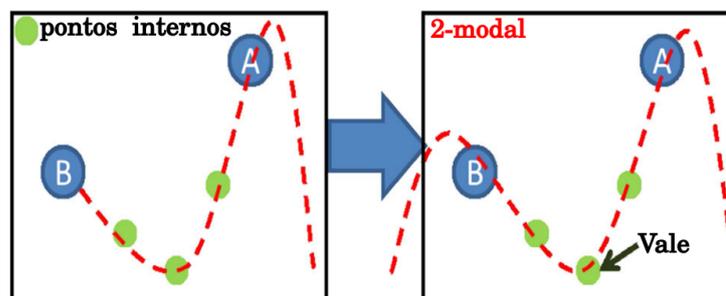
Algoritmo 6 *Nearest-better Clustering* (NBC)

- 1: Computa a distância entre todos os indivíduos
 - 2: Cria um grafo em que cada nó representa um indivíduo
 - 3: **para** $i \leftarrow 1$ **até** n **faça** \triangleright *spanning tree*
 - 4: Procura o melhor indivíduo mais próximo do indivíduo i e cria uma aresta entre eles
 - 5: **fim para**
 - 6: Exclui as arestas com tamanho $> (\phi \times \text{media}(\text{tamanhoTodasArestas}))$
 - 7: Cria os *clusters* com os componentes que permaneceram conectados
-

Após a separação dos indivíduos, são utilizados no máximo $nich_{max}$ nichos para serem-lhes aplicados o CMA-ES e geradas as proles de cada população. Estas são agrupadas em uma nova população, sendo novamente separadas através do *Nearest-better Clustering* (NBC). Com isso, são selecionados no máximo $nich_{max}$ sobreviventes para a próxima geração e o processo se repete até que um critério de parada seja atingido. Para preservar a população existente, μ indivíduos de cada agrupamento (*cluster*) permanecem em suas respectivas populações. Os valores dos parâmetros do CMA-ES a serem utilizados em cada nicho $nich_i^{(g)}$ são escolhidos conforme a configuração de $nich_j^{(g-1)}$ que gerou a melhor prole de $nich_i^{(g)}$. Onde o índice em subscrito é a identificação do nicho.

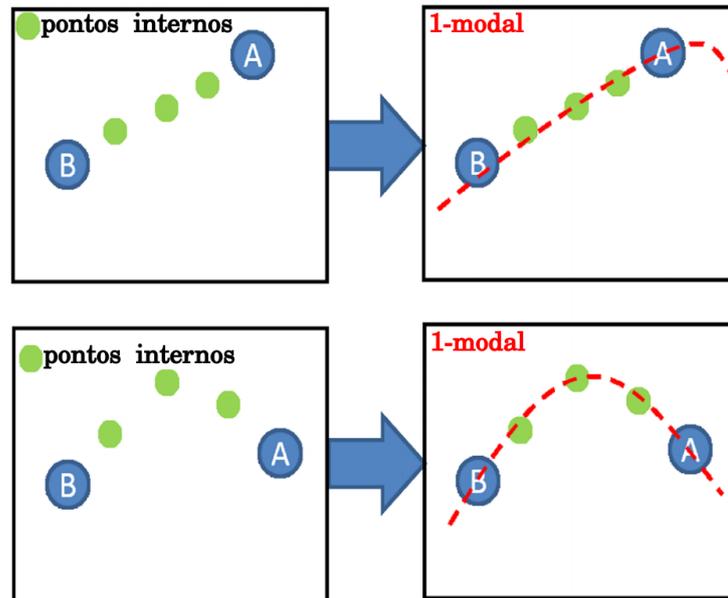
Ainda com vistas a tratar problemas de otimização mono-objetivo em ambiente multimodal, em Xu, Iizuka e Yamamoto (2017) os autores propõem utilizar um *Attraction Basin Sphere Estimation* (ABSE) (XU et al., 2013) com o CMA-ES. ABSE é um método que busca estimar a bacia esférica de atração de um determinado ponto, do inglês *Attraction Basin Sphere* (ABS), o qual tem o objetivo de detectar a multimodalidade do ambiente criando pontos intermediários entre um par de soluções (sejam A e B), identificando se há um vale entre elas. Caso exista este vale, diz-se que A e B constituem um par 2-modal (Figura 8), caso contrário este par é 1-modal (Figura 9).

Figura 8 – Dois pontos detectados como um par 2-modal



Fonte: Adaptado de Xu, Iizuka e Yamamoto (2017).

Figura 9 – Dois pontos detectados como um par 1-modal



Fonte: Adaptado de Xu, Iizuka e Yamamoto (2017).

Desta forma, realizando-se todas as análises pareadas possíveis, são armazenados para cada nodo o raio de vizinhança restrita (em que é possível abranger todos os indivíduos que compartilham daquela mesma região da superfície delimitada por vales) e o menor raio restrito aos vales (sendo a menor distância entre ele e o vale mais próximo). Assim, o raio de competência de cada nodo é o menor valor entre o raio de vizinhança restrita e o raio restrito aos vales.

Com isso Xu, Iizuka e Yamamoto (2017) criaram o algoritmo denominado *ABSE-CMA-ES* que tem por objetivo principal construir os nichos a cada geração, sendo-lhes aplicados o CMA-ES com seus respectivos hiperparâmetros. Os resultados obtidos no trabalho sugerem que o algoritmo proposto obtém melhor desempenho que *Attraction Basin Sphere Genetic Algorithm* (ABSGA) (XU et al., 2013), *DE/rand/2¹* (EPITROPAKIS; PLAGIANAKOS; VRAHATIS, 2011), CMA-ES com memória e *Niching Variable Mesh Optimization* (NMVO) (MOLINA et al., 2013).

Uma outra forma de evitar a convergência prematura do CMA-ES, é inserir imigrantes ao processo evolucionário. Esta técnica foi utilizada em Hansen (2011), onde o método desenvolvido realiza a inserção destes imigrantes que podem ser utilizados, inclusive, para substituir soluções ineficazes. As principais motivações apresentadas são: o CMA-ES gera soluções de forma simétrica, o que na maioria dos casos não representa melhores indivíduos (antes do processo de seleção); e bons imigrantes podem conduzir o algoritmo a locais promissores não contemplados pela população atual. Além disso, se o indivíduo inserido for ruim, não provocaria mudança significativa no CMA-ES, uma

¹ Esta notação é utilizada em DE (Evolução Diferencial) (EIBEN; SMITH, 2015), onde o primeiro termo indica o algoritmo, o segundo indica o vetor base e o terceiro o número de diferenças vetoriais. Podendo ainda ter mais um termo que indicaria o tipo de cruzamento.

vez que este se utiliza das melhores soluções para o processo de adaptação da matriz de covariância. Ainda em Hansen (2011) são listadas algumas opções de como gerar um imigrante:

- método gradiente;
- através de uma busca local, o que configuraria a construção de um algoritmo métrico;
- uma solução reparada, devido sua infactibilidade;
- uma solução ótima, obtida através de um outro modelo, construído com soluções já conhecidas;
- a melhor solução já encontrada até o momento;
- obtida através de qualquer algoritmo executado em paralelo ao CMA-ES

Na proposta apresentada, os imigrantes, gerados através da distribuição $\mathbf{1} + 10^{-4} \times \mathcal{N}(\mathbf{0}, \mathbf{I})$, substituem indivíduos da geração atual. Foram realizados testes com as funções: *Sphere* (KARABOGA; BASTURK, 2008) e *Rosenbrock* (POWELL, 1964). Em ambas houve redução da quantidade de avaliações de função, onde para a segunda foram obtidas 600 e 5000 avaliações de função com e sem a inserção de imigrantes, respectivamente, em espaço de busca $\in \mathbb{R}^{10}$. Em Hansen (2014) é demonstrado que utilizar o método gradiente no CMA-ES pode reduzir o número de iterações em funções unimodais.

Uma outra forma de inserir diversidade na população, para evitar a convergência prematura, é provocar perturbações aos indivíduos desta. Este método é utilizado no modelo chamado de *Covariance Matrix Adaptation Evolution Strategy with Directed Target to Best Perturbation Scheme* (CMA-ES-DTBP) (SINGH; PANIGRAHI; ABHYANKAR, 2013). Neste algoritmo, a população é gerada conforme o CMA-ES originalmente implementado e γ indivíduos (Equação 2.34) são escolhidos para serem perturbados conforme o Algoritmo 7, onde: $\mathbf{x}_{1:\lambda}^{(g)}$ é o melhor indivíduo da população; $i = \{1, \dots, \gamma\}$; o vetor aleatório $\mathbf{v}_i^{(g)} = \{v_1, \dots, v_D\}$, sendo $v_i \in \mathcal{N}(0, 1), \forall i \in \{1, \dots, D\}$; e $\hat{\mathbf{v}}_i^{(g)}$ é o vetor unitário obtido com $\mathbf{v}_i^{(g)} / \|\mathbf{v}_i^{(g)}\|$.

$$\gamma = \lceil \ln \{ |D \times \lambda \times \mathcal{N}(0, 1)| \} \rceil, \quad (2.34)$$

onde D é a dimensão do espaço de busca e λ é a quantidade de proles geradas a cada geração.

Algoritmo 7 *Directed Target to Best Perturbation* (DTBP)

- 1: $\mathbf{w}_i^{(g)} \leftarrow \mathbf{x}_{1:\lambda}^{(g)} - \mathbf{x}_i^{(g)}$
 - 2: $\mathbf{x}_i^{(g)} \leftarrow \mathbf{x}_i^{(g)} + \mathbf{w}_i^{(g)} \times \hat{\mathbf{v}}_i^{(g)}$
-

Dado que as funções objetivo podem possuir características distintas, podendo ser preferível exploração ou exploração em determinados momentos, utilizar o CMA-ES juntamente com outro método poderia ser uma relevante alternativa para otimizar funções multimodais. Baseando-se nesta suposição, (HE; ZHOU, 2018) apresenta o algoritmo memético chamado de *Differential Evolution variant with Covariance Matrix Adaptation Evolution Strategy* (DECMSA), em que os autores buscam uma harmonia entre as principais características de cada algoritmo utilizado: a boa exploração do DE com a capacidade de exploração do CMA-ES. No trabalho, os autores, utilizam uma nova técnica de mutação, denominada *DE/current - to - better/1*, a qual combina diferentes características do DE e CMA-ES. Nesta proposta, para cada indivíduo $\mathbf{x}_i^{(g)}$ são gerados n indivíduos a partir de uma distribuição Normal $\mathcal{N}(\mathbf{m}^{(g)}, \mathbf{C}^{(g)})$, onde a mutação procede conforme a Equação (2.35) para guiar a busca do DE. O controle do passo de mutação, proposto no CMA-ES original, é desconsiderado devido a aplicação da mutação do DE.

$$\mathbf{v}_i^{(g)} = \mathbf{x}_i^{(g)} + K_i^{(g)} (\mathbf{x}_{pbetter}^{(g)} - \mathbf{x}_i^{(g)}) + F_i^{(g)} (\mathbf{x}_{r1}^{(g)} - \hat{\mathbf{x}}_{r2}^{(g)}) \quad , \quad (2.35)$$

onde $\mathbf{x}_{pbetter}^{(g)}$ é a melhor solução gerada; $\mathbf{x}_{r1}^{(g)}$ é uma solução aleatoriamente selecionada, da geração g ; $\hat{\mathbf{x}}_{r2}^{(g)}$ é uma solução aleatoriamente escolhida, do conjunto formado pela união entre a população atual e um arquivo externo, o qual mantém as m soluções que não sobreviveram; $K_i^{(g)}$ e $F_i^{(g)}$ são parâmetros que são ajustados de forma diferente. A justificativa dada pelos autores para que não possuam o mesmo ajuste é que para uma função unimodal deseja-se que o valor de $K_i^{(g)}$ seja elevado, acelerando a convergência, e que o valor de $F_i^{(g)}$ seja nulo, não produzindo diversidade na população. Em contrapartida, em funções multimodais deseja-se o comportamento contrário.

Também é acrescida uma técnica de busca local, que é inicializada quando o critério de convergência descrito na Equação (2.36) é atingido, o qual apenas é acionado quando o processo evolucionário atinge 90% da quantidade máxima de avaliações de função permitida. Logo, a busca local consome aproximadamente 10% do total de avaliações de função. Os autores sugerem que podem ser realizadas várias buscas locais, contudo percebe-se que isso pode implicar na não convergência de uma região promissora.

$$\|\mathbf{C}^g\|_2 \leq \frac{1}{2} \times \max \{ub_j - lb_j | j = 1, \dots, D\} \quad , \quad (2.36)$$

onde $\|\mathbf{C}^g\|_2$ é a norma espectral da matriz de covariância \mathbf{C} ; ub_j e lb_j são os limites máximos e mínimos, respectivamente; e D representa a quantidade de dimensões do espaço de busca.

Além dos trabalhos apresentados nesta Seção, que têm por objetivo evitar a convergência prematura do CMA-ES em ambientes multimodais, existem estudos que sugerem modificações na forma de adaptar a matriz de covariância populacional, seja com o objetivo de uma melhor adaptação (JASTREBSKI; ARNOLD, 2006; LOSHCHILOV et al., 2014) ou para reduzir o custo computacional do algoritmo (ROS; HANSEN, 2008).

Todavia, o modelo utilizado como base para a construção do algoritmo proposto neste trabalho é o CMA-ES em sua concepção original (HANSEN, 2016), para que seja possível analisar a performance da nova proposta sem uma severa modificação na estrutura original do CMA-ES. Também há trabalhos que sugerem outras formas de adaptar o passo de mutação (SCHWEFEL, 1995; HANSEN; OSTERMEIER, 2001; HANSEN; ATAMNA; AUGER, 2014), porém apesar de influenciar no processo evolucionário, a adaptação do passo de mutação é independente do método de adaptação da matriz de covariância, conforme dito em Hansen (2016), e portanto não é o objetivo deste trabalho.

2.3 DISCUSSÃO DOS MÉTODOS APRESENTADOS

A partir do entendimento sobre a dinâmica do CMA-ES, descrita ao longo da Seção 2.1, e dos modelos apresentados na Seção 2.2, percebe-se que os esforços para evitar que o CMA-ES convirja prematuramente estão baseados em duas ações: provocar diversidade e explorar regiões distintas do espaço de busca.

Nas abordagens baseadas em reinicialização do processo evolucionário, o que se deseja é possibilitar que o algoritmo explore regiões distintas do espaço de busca, a cada reinicialização, muitas vezes sem utilizar o conhecimento sobre o que já foi explorado. Pode-se perceber que neste tipo de abordagem o sucesso está condicionado a uma boa escolha do momento de reinicializar o processo e no tamanho da população a ser utilizada.

Já nas abordagens baseadas em nichos, esta exploração é realizada desde o início, uma vez que são utilizadas subpopulações distintas. Contudo, deve-se ressaltar que isso implica num maior consumo de avaliações de função, deste o início do processo evolucionário, dependendo da quantidade de nichos e do tamanho de cada subpopulação.

Quanto à abordagens que utilizam inserção de imigrantes, deve-se analisar qual será(ão) o(s) indivíduo(s) a ser(em) substituído(s), e como o(s) imigrante(s) será(ão) gerado(s) (aleatoriamente, através de um histórico, etc). Esta última pode, inclusive, necessitar a inclusão de um outro método, como o gradiente, para esta finalidade. Um ponto positivo a ser levado em consideração neste tipo de abordagem é que se o imigrante for ruim ele não influenciará no CMA-ES, uma vez que a própria dinâmica do processo se encarregará de retirá-lo da população. Porém, se ele tiver uma boa aptidão, guiará a população para uma região promissora.

Por fim, os algoritmo meméticos baseados no CMA-ES podem ser uma boa alternativa para lidar com problemas distintos, desde que seja realizada uma boa escolha do método que será integrado ao processo evolucionário, observado-se sua dinâmica frente ao objetivo que se deseja alcançar com sua inclusão.

Diante destes fatos, no Capítulo 3 pode ser visto um estudo quantitativo sobre o comportamento do CMA-ES, bem como algumas sugestões iniciais para fazer com que seu desempenho em funções multimodais seja melhorado, baseando-se no que já foi

discutido. Este estudo preliminar subsidiará a construção do modelo proposto nesta dissertação, o qual encontra-se descrito ao final daquele Capítulo.

3 LOCAL CMA-ES GUIDED BY DIFFERENTIAL EVOLUTION (L-CMA-DE)

Ficou notório no Capítulo 2 que uma das principais limitações do CMA-ES é que devido a sua concepção com características de um método de busca local que pode, com frequência, levar a retenção do processo evolucionário simulado a um ótimo local. Esta limitação tende a se agravar quando a paisagem do espaço de busca é multimodal, tal como a função *Rastrigin* (MÜHLENBEIN; SCHOMISCH; BORN, 1991). Ao longo dos últimos anos, algumas abordagens foram propostas para tratar tal limitação, como visto na Seção 2.2.

Como o interesse da presente dissertação é melhorar o desempenho do CMA-ES em ambientes multimodais, toda a pesquisa foi direcionada com este objetivo. Desta forma, foi observado um avanço neste quesito nos algoritmos apresentados na Seção 2.2, proporcionado por métodos que tem por objetivo explorar o espaço de busca, minimizando a dependência da inicialização do CMA-ES e evitando a convergência prematura. Contudo, ainda permanece a limitação de que o CMA-ES necessita de uma população grande para que se alcance bons resultados em funções objetivo difíceis, além de não ser fácil identificar o momento no processo evolucionário em que se deva realizar a intervenção (LOSHCHILOV; SCHOENAUER; SEBAG, 2012). Além disso, alguns modelos propostos trazerem consigo alguns parâmetros cujos ajustes estão diretamente correlacionados com o desempenho do algoritmo (XU; IIZUKA; YAMAMOTO, 2017).

Apesar de todo o esforço despendido, uma característica não muda dentre as pesquisas expostas no presente trabalho: o CMA-ES continua com elevada dependência do seu ponto inicial (ou de reinicialização), pois é a partir daquele local que todo o processo começa a ser moldado e se estiver próximo a mínimos locais, as chances do algoritmo ficar preso em algum destes é elevada. É frente a esta característica que o presente trabalho está focado e os estudos apresentados nas Seções a seguir darão suporte à construção do método proposto na Seção 3.5.

3.1 ANÁLISE DA EVOLUÇÃO DO CMA-ES PARA PROBLEMAS MONO-OBJETIVO

Como visto na Seção 2.1, o CMA-ES é uma Estratégia Evolucionária bem elaborada que alcança expressivo desempenho em ambientes unimodais devido a forma como a matriz de covariância é adaptada: maximizando a probabilidade de gerarem-se indivíduos na direção promissora. Contudo, seu desempenho em ambientes multimodais é significativamente afetado, podendo convergir prematuramente para mínimos locais (HANSEN; KERN, 2004). Buscando atenuar esta limitação, alguns algoritmos surgiram (ver Seção 2.2). Duas principais estratégias foram adotadas, sejam elas: reinicializar o processo evolucionário, caso seja percebido estagnação do algoritmo; e criação de nichos.

Na primeira, buscou-se inicialmente incrementar a população a cada reinício,

através do IPOP-CMA-ES (AUGER; HANSEN, 2005), com a esperança de melhor explorar o ambiente. Este método possibilitou, além da exploração de regiões durante o processo de reinicialização, uma melhor observação entorno do ponto médio gerador dos indivíduos, todavia ele ainda continuou a ficar preso em mínimos locais, pois seu sucesso está extremamente ligado à inicialização/reinicialização em regiões promissoras, além de trazer consigo um outro ponto desfavorável ao desempenho que é o aumento da quantidade de avaliações de função. Para lidar com o aumento das avaliações, em 2006 foi proposta uma nova forma para adaptar a matriz de covariância, o método foi batizado por *Active Covariance Matrix Adaptation Evolution Strategy* (*Active-CMA-ES*) (JASTREBSKI; ARNOLD, 2006). Esta adaptação busca fazer com que a medida em que o CMA-ES seja executado, a matriz de covariância seja modificada de modo a potencializar as chances de serem gerados indivíduos em direções promissoras. Contudo, esta proposta de adaptação não reduz a probabilidade de serem gerados indivíduos em direções que não sejam promissoras, efeito colateral do *Active-CMA-ES*. Desta feita surge o IPOP-aCMA-ES, que além de incrementar a população a cada reinicialização, também utiliza a informação sobre direções não promissoras para reduzir a probabilidade de buscas naquela região, contudo a população continua sendo aumentada a cada reinicialização, o que conseqüentemente aumenta a quantidade de avaliações de função. Para evitar ou amenizar este acréscimo da quantidade de avaliações, BIPOP-CMA-ES (HANSEN, 2009) pode adaptar o tamanho da população (aumentar ou diminuir) dependendo da qualidade das soluções geradas por cada uma das estratégias de adaptação. Este algoritmo também ganhou uma versão em combinação com o $(\mu/\mu_w, \lambda)$ -aCMA-ES, chamada BIPOP-aCMA-ES, além de também haver uma outra extensão que decremente o passo de mutação a cada reinício, denominada NBIPOP-aCMA-ES (LOSHCHILOV, 2013) que ganhou a competição da *IEEE Congress on Evolutionary Computation* (CEC) no ano de 2013, o que sugere seu valor para a comunidade científica. Contudo, como pode ser visto em (LOSHCHILOV; SCHOENAUER; SEBAG, 2012), as técnicas de reinicialização apresentadas acima ainda possuem dependência do local de (re-)início do algoritmo.

Buscando diminuir a dependência do ponto inicial da busca, surgiram abordagens que empregam nichos. Em um contexto geral, estas buscam criar subpopulações em regiões distintas, para que assim seja possível melhor explorar o ambiente. Como cada proposta tem seu custo agregado, pode-se notar que ao serem criados e evoluídos nichos separadamente, o número de avaliações de função tende a ser alto, o que pode ser um problema se esse número for considerado como um dos critérios de parada do algoritmo, entretanto para tratar problemas do mundo real com tempo para processamento, pode ser uma estratégia muito útil.

3.2 COMPORTAMENTO DA MATRIZ DE COVARIÂNCIA

Com vistas às diversas modificações aplicadas ao CMA-ES (algumas delas abordadas no Capítulo 2), nesta seção estão expostas algumas sugestões de mudanças ao algoritmo original, bem como as provas experimentais de seus impactos. Para escolher e fundamentar as propostas implementadas, buscou-se entender o comportamento da matriz de covariância ao longo das gerações. Analisado o que foi apresentado sobre o CMA-ES, espera-se que durante o processo evolucionário, a busca em torno do vetor médio seja reduzida em amplitude à medida que o algoritmo for se aproximando do ótimo local ou global. Procurando comprovar esta esperança, foram realizados 31 testes independentes, utilizando a função Esfera e espaço de busca $\in \mathbb{R}^{10}$, e computada a esparsidade (S) das matrizes de covariâncias (\mathbf{C}) a cada 100 gerações através da Equação (3.1).

$$S = \frac{|-lim \leq c_{i,j} \leq lim|}{|\mathbf{C}|}, \quad \forall_{i,j}, \quad (3.1)$$

onde $|\cdot|$ é o cardinal; lim é a distância máxima partindo-se de 0, sendo 0,0005 o valor empiricamente escolhido, uma vez que os valores de entrada da matriz $\mathbf{C} \in [-5 \times 10^{-4}, 5 \times 10^{-4}]$ produzem indivíduos bastante semelhantes; e $c_{i,j}$ o elemento localizado na i -ésima linha e j -ésima coluna da matriz \mathbf{C} . Logo, $S \in [0, 1]$ e representa a porcentagem de valores de \mathbf{C} que estão dentro do intervalo estipulado. Os resultados obtidos podem ser vistos na Tabela 1.

Tabela 1 – Esparsidade da matriz de covariância ao longo das gerações (Função Esfera)

Teste	Geração					
	100	200	300	400	500	600
01	0,02	0,06	0,19	0,36	0,49	-
02	0,02	0,10	0,25	0,32	0,55	-
03	0,06	0,10	0,17	0,28	0,49	0,68
04	0,00	0,04	0,15	0,36	0,47	-
05	0,04	0,08	0,21	0,36	0,43	0,64
06	0,04	0,06	0,19	0,44	0,45	-
07	0,00	0,08	0,15	0,38	0,59	0,70
08	0,00	0,04	0,25	0,34	0,41	0,66
09	0,00	0,06	0,17	0,34	0,51	-
10	0,04	0,08	0,19	0,38	0,49	0,68
11	0,00	0,04	0,21	0,32	0,53	-
12	0,00	0,08	0,21	0,42	0,49	-
13	0,00	0,08	0,11	0,34	0,43	-
14	0,02	0,04	0,11	0,26	0,43	0,64
15	0,02	0,10	0,29	0,48	0,47	-
16	0,00	0,04	0,15	0,28	0,55	-
17	0,04	0,08	0,21	0,38	0,47	-
18	0,02	0,08	0,23	0,40	0,49	-
19	0,04	0,10	0,23	0,38	0,45	0,70
20	0,02	0,10	0,13	0,34	0,49	0,72
21	0,02	0,06	0,21	0,34	0,49	-
22	0,02	0,04	0,19	0,38	0,49	-
23	0,00	0,04	0,21	0,26	0,45	-
24	0,04	0,08	0,14	0,36	0,43	0,60
25	0,02	0,04	0,21	0,38	0,45	-
26	0,00	0,04	0,23	0,40	0,59	-
27	0,04	0,08	0,15	0,28	0,45	0,66
28	0,00	0,04	0,17	0,28	0,53	0,60
29	0,02	0,08	0,25	0,28	0,51	0,66
30	0,02	0,18	0,19	0,30	0,55	-
31	0,00	0,12	0,23	0,32	0,39	0,62

Fonte: Próprio Autor

A função Esfera foi utilizada por ser uma função mono-modal de fácil convergência para o CMA-ES, sendo 600 o número máximo de gerações apresentado porque em nenhum dos testes o algoritmo chega a 700 gerações para a função objetivo escolhida. Os

testes em que não constam valores na coluna de 600 gerações estão assim representados porque o algoritmo convergiu para o ótimo global antes desta quantidade de gerações. Através da Tabela 1 observa-se que o nível de esparsidade da matriz \mathbf{C} aumenta à medida que o algoritmo converge para o mínimo. Este comportamento é ocasionado devido à proximidade entre as μ melhores soluções de uma mesma geração e de gerações consecutivas, sugerindo que ao final do processo evolucionário espera-se que o melhor indivíduo seja o vetor médio da última geração. Buscando verificar se ocorre este mesmo comportamento em funções multimodais, o mesmo teste foi realizado com a função 22 (*Composition Function 2*) da CEC-2013 (LIANG et al., 2013). Os resultados deste último experimento podem ser vistos na Tabela 2, demonstrando que o comportamento observado em uma função unimodal se repete em funções multimodais, havendo nesta última uma convergência para um ótimo local.

Tabela 2 – Esparsidade da matriz de covariância ao longo das gerações (Função 22 da CEC-2013)

Teste	Geração											
	100	500	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
01	0.06	0.38	0.96	1	1	1	1	1	1	1	1	1
02	0.02	0.42	0.51	0.97	1	1	1	1	1	1	1	1
03	0	0.24	0.97	1	1	1	1	1	1	1	1	1
04	0.02	0.16	0.94	1	1	1	1	1	1	1	1	1
05	0	0.34	0.42	0.99	1	1	1	1	1	1	1	1
06	0	0.19	0.48	0.99	1	1	1	1	1	1	1	1
07	0	0.36	0.53	0.94	1	1	1	1	1	1	1	1
08	0.04	0.38	0.99	1	1	1	1	1	1	1	1	1
09	0.02	0.18	0.95	1	1	1	1	1	1	1	1	1
10	0	0.34	0.99	1	1	1	1	1	1	1	1	1
11	0	0.24	0.96	1	1	1	1	1	1	1	1	1
12	0	0.19	0.33	1	1	1	1	1	1	1	1	1
13	0.02	0.32	0.44	1	1	1	1	1	1	1	1	1
14	0	0.25	0.41	1	1	1	1	1	1	1	1	1
15	0.02	0.19	0.37	1	1	1	1	1	1	1	1	1
16	0.02	0.21	0.33	1	1	1	1	1	1	1	1	1
17	0.04	0.36	0.38	1	1	1	1	1	1	1	1	1
18	0	0.04	0.65	0.64	0.64	0.64	0.7	0.82	0.94	0.96	0.96	0.97
19	0.02	0.36	0.4	1	1	1	1	1	1	1	1	1
20	0	0.32	0.98	1	1	1	1	1	1	1	1	1
21	0.02	0.22	0.98	1	1	1	1	1	1	1	1	1
22	0	0.25	0.33	1	1	1	1	1	1	1	1	1
23	0	0.44	0.99	1	1	1	1	1	1	1	1	1
24	0	0.23	0.25	1	1	1	1	1	1	1	1	1
25	0.02	0.44	0.98	1	1	1	1	1	1	1	1	1
26	0.02	0.19	0.61	1	1	1	1	1	1	1	1	1
27	0	0.29	0.66	0.9	1	1	1	1	1	1	1	1
28	0	0.41	0.51	0.83	1	1	1	1	1	1	1	1
29	0.04	0.44	0.89	1	1	1	1	1	1	1	1	1
30	0.02	0.34	0.97	1	1	1	1	1	1	1	1	1
31	0.02	0.24	0.92	1	1	1	1	1	1	1	1	1

Fonte: Próprio Autor

3.3 INSERÇÃO DE DIVERSIDADE

De acordo com o experimento realizado na Seção 3.2, ao passo que o CMA-ES evolui, a população perde diversidade. Além disso, conforme as informações expostas ao longo da Seção 2.2, nota-se que induzir diversidade durante o processo evolucionário pode resultar em melhores soluções. Diante dessa afirmativa, a primeira sugestão foi causar diversidade utilizando o ponto médio entre os μ melhores indivíduos da geração de uma

dada população (chamada de principal), para que a partir deste ponto seja possível aplicar o CMA-ES para gerar e evoluir uma outra população (chamada de alternativa), baseando-se em diversidade. Para isso, a função de aptidão da população alternativa deixa de ser a originalmente concebida pelo processo evolucionário da população principal e passa a ser uma dada medida de distância¹, onde aqueles indivíduos que apresentarem a maior distância média (Equação 3.2), passarão a guiar o processo alternativo.

$$\bar{d}_i = \frac{1}{\lambda} \sum_{j=1}^{\lambda} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \quad , \quad \forall i \neq j, \quad (3.2)$$

onde λ é o tamanho da população e $\|\cdot\|_2^2$ representa a distância euclidiana.

Desta forma, espera-se que a população alternativa realize uma exploração em torno dos μ melhores indivíduos da população principal, com a motivação de que pode haver alguma(s) melhor(es) região(ões) nas proximidades daquela ocupada por estes indivíduos. Após um determinado critério de parada ser atingido², um percentual (30%) dos piores indivíduos da população principal é substituído pelo mesmo percentual dos melhores da população alternativa, caso estes apresentem-se mais aptos. Após isto, a população principal é novamente ranqueada e o CMA-ES continua o processo de adaptação conforme método originalmente concebido.

Para verificar o desempenho da proposta apresentada no parágrafo anterior, foram executados 10 testes para cada uma das 28 funções objetivo (com espaço de busca em \mathbb{R}^{10}) constantes em Liang et al. (2013). Para identificar se os resultados obtidos forneciam algum indicativo de melhora das soluções finais, foi executado o teste de *Mann-Whitney-Wilcoxon* na plataforma *R* (VENABLES; SMITH; TEAM, 2017) entre o CMA-ES original e esta nova proposta. Para os testes realizados foi considerado nível de confiança de 95% ($\alpha = 0,05$) e os resultados estão apresentados na Tabela 3. Os valores em negrito indicam que em algum(ns) dos testes aquele algoritmo alcançou o ótimo global. O termo "NA" na coluna "Valor-P" indica que os valores foram idênticos em todos os testes de ambos algoritmos. Para que os resultados obtidos fossem considerados estatisticamente diferentes, seria necessário que o Valor-P apresentasse valores abaixo de α . Portanto, pode-se observar que em nenhuma das funções objetivo os algoritmos apresentaram diferença estatística, ou seja, houve um maior custo computacional com a nova proposta, contudo sem melhorias nos resultados de aptidão.

¹ Distância euclidiana

² 20 gerações

Tabela 3 – *Mann-Whitney-Wilcoxon Test* (CMA-ES \times CMA-ES - diversidade com os μ melhores)

Função Objetivo	CMA-ES (Mediana)	CMA-ES - diversidade com μ melhores (Mediana)	Valor-P
01	-1400,000000	-1400,000000	NA
02	-1300,000000	-1300,000000	0,3681000
03	-1200,000000	-1200,000000	0,8953000
04	-1100,000000	-1100,000000	0,3681000
05	-1000,000000	-1000,000000	NA
06	5774,5250000	5341,9850000	0,2229000
07	-687,9080000	-30,7899000	0,1390000
08	-678,8905000	-678,5815000	0,3930000
09	-585,5725000	-587,0825000	0,5288000
10	-499,9830000	-499,9900000	0,2833000
11	-384,5785000	-381,5935000	0,9698000
12	-43,3078900	141,7395000	0,1857000
13	-160,0540000	-173,5180000	0,2799000
14	1560,9400000	1981,3300000	0,2799000
15	1565,5750000	1792,1700000	0,3150000
16	200,1320000	200,1740000	0,4725000
17	1092,2250000	1034,2650000	0,5288000
18	1257,2550000	1351,3700000	1,0000000
19	501,2695000	500,9455000	0,1431000
20	605,0000000	604,9975000	0,7456000
21	1100,1900000	1100,1900000	0,3681000
22	3101,6500000	3002,3250000	0,9118000
23	3546,2350000	3266,5900000	0,3150000
24	1258,5350000	1400,6250000	0,1986000
25	1358,3250000	1349,3300000	0,7394000
26	1535,6550000	1542,9150000	0,7959000
27	1864,4300000	1869,7800000	0,7591000
28	1700,0000000	1700,0000000	0,1982000

Fonte: Próprio Autor

Foi observado que a forma de evolução da população alternativa, utilizada no experimento descrito acima, faz com que os indivíduos que devem guiar o processo estejam distantes entre si e que o centro de massa fique bem próximo ao obtido pelo processo principal, não trazendo significantes benefícios ao processo evolucionário. Diante dessa análise, a evolução da população alternativa foi modificada para que aqueles indivíduos que estivessem mais distantes do ponto médio da população principal (desconsiderando a

distância entre eles) guiassem a evolução do processo alternativo, gerando um novo centro de massa. O processo se repete até que o número máximo de gerações seja atingido. Desta forma, se forem gerados indivíduos em regiões distantes do centroide e próximos entre si, estes deslocarão o centróide, explotando em regiões próximas aos μ melhores indivíduos da população principal. O resultado dos testes podem ser vistos na Tabela 4.

Tabela 4 – *Mann-Whitney-Wilcoxon Test* (CMA-ES \times CMA-ES - diversidade com os μ melhores - aptidão do processo alternativo sendo a distância para o centroide)

Função Objetivo	CMA-ES (Mediana)	CMA-ES - diversidade com μ melhores (Mediana)	Valor-P
01	-1400,000000	-1400,000000	NA
02	-1300,000000	-1300,000000	NA
03	-1200,000000	47433949400,000000	0,5310000
04	-1100,000000	-1100,000000	0,3681000
05	-1000,000000	-1000,000000	0,3681000
06	5774,5250000	5774,5250000	0,9044000
07	-687,9080000	-737,6740000	0,5421000
08	-678,8905000	-678,8355000	0,7394000
09	-585,5725000	-587,0050000	0,9705000
10	-499,9830000	-499,9865000	0,6211000
11	-384,5785000	-373,1360000	0,1730000
12	-43,3078900	61,1560000	0,5966000
13	-160,0540000	-163,1330000	0,7959000
14	1560,9400000	1643,2500000	0,4813000
15	1565,5750000	2033,5350000	0,0432600
16	200,1320000	200,1365000	0,9698000
17	1092,2250000	1192,5400000	0,9118000
18	1257,2550000	1123,2150000	0,3150000
19	501,2695000	501,2395000	0,6842000
20	605,0000000	605,0000000	0,3278000
21	1100,1900000	1100,1900000	1,0000000
22	3101,6500000	3397,1050000	0,2176000
23	3546,2350000	3546,2350000	0,1431000
24	1258,5350000	1497,6300000	0,4960000
25	1358,3250000	1346,4200000	0,8501000
26	1535,6550000	1535,1500000	0,4958000
27	1864,4300000	2052,2400000	0,6747000
28	1700,0000000	1700,0000000	0,6257000

Fonte: Próprio Autor

Analisando a Tabela 4 é possível observar que não houve uma significativa mudança, em relação aos resultados da Tabela 3, com apenas o resultado da função 15 apresentando-se estatisticamente diferente entre os algoritmos comparados, todavia com uma maior mediana dos resultados para a proposta implementada.

Uma outra alternativa para provocar diversidade foi testada: foi mantida uma população alternativa que a cada 100 gerações substituía 30% de seus piores indivíduos pelos melhores da população principal, caso estes apresentassem melhor aptidão que aqueles, e evoluía por 20 épocas. A cada 50 gerações do processo principal, foi verificado se havia no máximo 30% de indivíduos deste que fossem piores que os melhores do processo alternativo, em isto ocorrendo a população principal os substituía pelos imigrantes. Os resultados dos testes realizados podem ser vistos na Tabela 5.

Tabela 5 – *Mann-Whitney-Wilcoxon Test* (CMA-ES × CMA-ES - aplicando diversidade mútua entre a população principal e a alternativa)

Função Objetivo	CMA-ES (Mediana)	CMA-ES com diversidade mútua (Mediana)	Valor-P
01	-1400,000000	-1400,000000	NA
02	-1300,000000	-1300,000000	NA
03	-1200,000000	-1200,000000	1,000000
04	-1100,000000	-1100,000000	1,000000
05	-1000,000000	-1000,000000	NA
06	5774,525000	5733,390000	0,453200
07	-687,908000	-45,966100	1,000000
08	-678,890500	-678,668000	0,198600
09	-585,572500	-587,420000	0,970500
10	-499,983000	-499,990000	0,073020
11	-384,578500	-378,111000	0,405300
12	-43,307890	-131,855500	1,000000
13	-160,054000	-159,538500	1,000000
14	1560,940000	1587,505000	0,684200
15	1565,575000	2106,640000	0,190300
16	200,132000	200,137500	0,939700
17	1092,225000	764,309000	0,035460
18	1257,255000	937,601000	0,123000
19	501,269500	500,726500	0,105100
20	605,000000	605,000000	0,101200
21	1100,190000	1100,190000	1,000000
22	3101,650000	3272,300000	0,911800
23	3546,235000	3113,865000	0,123000
24	1258,535000	1266,025000	0,911800
25	1358,325000	1346,670000	1,000000
26	1535,655000	1523,945000	0,879800
27	1864,430000	1929,990000	1,000000
28	1700,000000	1700,000000	0,824500

Fonte: Próprio Autor

Ao analisar a Tabela 5 percebe-se que apenas na função 17 foi possível observar uma melhora estatisticamente relevante em comparação aos resultados obtidos com o CMA-ES original. Nas demais houve um acréscimo de esforço computacional, porém sem ganhos significativos nos resultados finais.

Sabendo-se que o subespaço explorado pelo CMA-ES está diretamente relacionado com a sua inicialização, possibilitando que outras regiões (que poderiam ser mais

promissoras) deixem de ser exploradas, uma alternativa seria realizar uma busca em regiões ainda não exploradas. Como não é trivial saber a região ainda não explorada, seja pelo não conhecimento desta região em problemas do mundo real ou pelo esforço computacional, então uma forma de provocar esta exploração é gerar γ indivíduos a depender de um determinado critério. Logo, foi realizado um estudo gerando-se 200 indivíduos através de distribuição Uniforme, dentro do limite espacial estabelecido pela função objetivo, sempre que o grau esparsidade da matriz de covariância da população atingisse um determinado valor. Em seguida o indivíduo mais apto, dentre os aleatoriamente gerados, é comparado com o vetor médio gerado pelo CMA-ES, permanecendo como vetor médio para a geração seguinte aquele que obtiver o melhor valor de aptidão.

Os resultados estão apresentados na Tabela 6, onde CMAESDivRand_200 é a nova proposta. As melhores médias, com 95% de nível de confiança, estão apresentadas em negrito. Como pode ser visto, os resultados obtidos são satisfatórios e corroboram com o conhecimento sobre o comportamento do CMA-ES, demonstrando que se faz necessário inserção de indivíduos diversos para que a população possam migrar de ótimos locais para regiões mais promissoras.

Tabela 6 – *Mann-Whitney-Wilcoxon Test* (CMA-ES × CMAESDivRand_200)

Número	CMA-ES				CMAESDivRand_200				D _{stat}
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
1	-1400.0000	-1229.6140	2694.0460	-1400.0000	-1400.0000	-1400.0000	0.0000	-1400.0000	0.3193
2	-1300.0000	56182833.0000	205780448.0000	-1300.0000	-1300.0000	-1300.0000	0.0000	-1300.0000	0.0000
3	-1200.0000	152668104494.0000	107205600000.0000	-1200.0000	-1200.0000	-1200.0000	0.0000	-1200.0000	0.0000
4	-1100.0000	219526.4000	2075844.0000	-1100.0000	-1100.0000	-1100.0000	0.0000	-1100.0000	0.0451
5	-1000.0000	5519.0880	59271.3800	-1000.0000	-1000.0000	-1000.0000	0.0000	-1000.0000	0.0831
6	5733.3900	5572.2310	6083.8080	-900.0000	5733.3900	3657.5120	3406.2250	-900.0000	0.0027
7	-45.9661	922174.3000	14565334.0000	-799.9930	-732.3310	6359312.0000	95488085.0000	-799.9990	0.0000
8	-678.7815	-678.9243	0.4445	-680.0000	-678.6770	-678.6965	0.1979	-680.0000	0.0000
9	-586.1260	-586.8611	4.0619	-599.0740	-591.0135	-591.3072	2.7712	-598.0240	0.0000
10	-499.9900	-478.3883	341.4719	-500.0000	-500.0000	-499.9972	0.0083	-500.0000	0.0000
11	-378.6085	-201.6104	285.8099	-397.0150	-379.1050	-359.5342	37.4713	-397.0150	0.0068
12	-96.5351	-18.1855	301.2322	-295.0250	-270.6485	-251.5433	41.5483	-298.0100	0.0000
13	-166.2430	-154.5047	35.6059	-198.0100	-157.5410	-145.8542	38.5312	-198.0100	0.0010
14	1703.0700	1692.0630	410.1419	514.1170	1545.0900	1555.4220	355.8894	646.4020	0.0003
15	1877.7200	1909.9010	482.0812	720.6310	1674.2600	1668.1190	377.3527	467.8520	0.0000
16	200.1250	200.1508	0.1249	200.0000	200.0860	200.1622	0.3810	200.0000	0.0000
17	979.3735	928.6360	496.0374	305.6250	374.6095	382.2389	58.2772	313.1420	0.0000
18	1017.9100	981.4880	523.4094	413.0170	430.2675	449.5460	42.1407	411.5180	0.0000
19	501.0060	19578.9700	176126.5000	500.3750	500.9485	501.0613	0.5718	500.2580	0.1004
20	605.0000	604.9044	0.3021	602.8020	604.4540	604.3963	0.3348	603.0820	0.0000
21	1100.1900	1077.7700	69.8957	800.0000	1100.1900	1075.7990	71.7096	800.0000	0.7004
22	3114.6800	3080.9470	422.8319	1820.2500	2852.8900	2820.4850	395.3648	1722.0400	0.0000
23	3225.3200	3226.8320	493.1024	1920.9400	2975.6450	2945.6680	351.9444	1767.3800	0.0000
24	1373.7700	1399.0210	154.8157	1106.5800	1230.2250	1224.2370	26.2282	1109.9700	0.0000
25	1345.0100	1359.1530	48.9874	1209.0000	1326.9700	1326.6540	12.6916	1294.1800	0.0000
26	1541.3200	1511.6360	121.2349	1303.9800	1514.3400	1459.2440	102.0939	1265.5600	0.0000
27	2039.5600	1908.7620	301.3010	1600.2700	1700.0000	1842.8790	200.0547	1454.4500	0.0019
28	1700.0000	2557.0160	1192.5380	1500.0000	1756.8500	1974.9900	328.0442	1500.0000	0.0244

Fonte: Próprio Autor

Visando gerar possíveis imigrantes de uma forma mais eficaz, foi implementada uma alternativa em que os indivíduos gerados aleatoriamente são evoluídos através de DE, para em seguida ser realizada a comparação entre o vetor médio da população

gerada pelo CMA-ES e o melhor indivíduo gerado pelo DE, os passos seguintes seguem conforme descrito no parágrafo anterior. *CMAES_DivRandDERepMean* foi o nome dado a esta proposta. Os resultados podem ser analisados na Tabela 7, onde foram executados 51 testes independentes com cada algoritmo, utilizando-se as funções da CEC-2013 com espaço de busca em 10 dimensões. Observa-se que os resultados apresentam-se com qualidade superior, utilizando o novo método. Esta ocorrência se deve ao poder de exploração do DE, conforme indicado em Zhou e Qiao (2019), fazendo com que seja possível evoluir as soluções geradas aleatoriamente, produzindo indivíduos mais aptos. Desta forma possibilita que o CMA-ES saia de um ótimo local e passe a explorar em uma região mais promissora.

Tabela 7 – *Mann-Whitney-Wilcoxon Test* (CMAES_DivRandDERepMean \times CMAES-DivRand_200)

Função	CMAES_DivRandDERepMean				CMAESDivRand_200				Melhor Aptidão	D. Valor
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão		
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	NaN
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	NaN
3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	NaN
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	NaN
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	NaN
6	0.3710	6.9795	10.3927	0.0000	0.0000	7.3890	9.7963	0.0000	0.0000	0.8265
7	35.5860	35.6562	31.1740	0.0020	57.4860	67.6095	68.5289	0.0010	0.0010	0.0447
8	20.3490	20.3419	0.1011	20.0000	20.3770	20.3456	0.1021	20.0000	20.0000	0.6880
9	5.3540	5.3436	1.4972	1.8930	8.5700	7.7137	2.4572	1.5280	1.5280	0.0000
10	0.0000	0.0011	0.0028	0.0000	0.0000	0.0010	0.0024	0.0000	0.0000	0.9419
11	19.8990	22.7085	12.8694	1.9900	22.8840	42.6073	39.1749	2.9850	2.9850	0.1835
12	16.9140	21.5573	16.4797	1.9900	30.8440	56.1854	49.7967	2.9850	2.9850	0.0020
13	42.9930	50.6833	29.3604	1.9900	42.1810	61.4183	48.3585	2.0030	2.0030	0.7178
14	1090.8960	1054.7990	413.4651	149.7266	1444.9400	1471.7760	403.0357	423.3890	423.3890	0.0000
15	1165.4800	1137.0200	363.8295	141.8370	1344.6800	1429.0870	430.5845	424.2210	424.2210	0.0037
16	0.0320	0.0330	0.0250	0.0000	0.0220	0.0238	0.0157	0.0000	0.0000	0.1023
17	43.2660	42.6717	18.9302	12.3960	35.4390	78.8508	74.4272	6.5570	6.5570	0.6442
18	29.6100	37.0358	22.0015	5.3360	22.6590	38.9314	44.5941	5.3930	5.3930	0.3972
19	0.8600	0.9419	0.4010	0.2680	0.8690	0.9681	0.4257	0.1900	0.1900	0.7506
20	3.6230	3.5257	0.3845	1.9740	4.2730	4.2323	0.4312	2.8470	2.8470	0.0000
21	400.1900	355.0584	100.6970	0.0000	400.1900	390.3749	41.2982	100.0000	100.0000	0.0525
22	929.9400	967.7448	414.0642	139.1170	1858.1800	1879.0230	338.1687	460.7900	460.7900	0.0000
23	1155.8300	1221.9390	345.2058	320.8300	1868.8900	1886.1470	465.4705	280.2500	280.2500	0.0000
24	219.2700	218.5533	5.2162	102.1300	225.9200	221.8522	21.2891	110.1300	110.1300	0.0000
25	219.4500	214.2808	20.5628	115.8000	223.6600	219.4690	17.7047	121.4700	121.4700	0.0038
26	200.0200	176.1916	38.7494	94.1900	190.9000	166.0704	40.2160	103.9800	103.9800	0.3430
27	471.3700	482.2657	110.7503	226.7100	538.7600	506.6527	138.4839	300.0200	300.0200	0.2464
28	300.0000	485.3153	243.8107	100.0000	300.0000	431.0404	284.7181	100.0000	100.0000	0.1701

Fonte: Próprio Autor

3.4 CMA-ES COM DIVISÃO DO ESPAÇO DE BUSCA

A divisão do espaço é bastante utilizada nos mais diversos campos da ciência, sobretudo na engenharia, através do Método dos Elementos Finitos (MEF), do inglês *Finite Element Method* (FEM) (DHATT et al., 2012), que é utilizado principalmente para modelar estruturas. Neste método o espaço é subdividido e procura-se uma função que

se aproxime da geometria de cada subespaço. Fica claro que quanto mais subespaços houver, maior a precisão do que poderá ser obtida a expensas do aumento do custo computacional. Conforme Azevedo (2003), a formulação do MEF requer a existência de uma equação integral, de forma que possibilite substituir o integral sobre um domínio complexo (de volume V) pelo somatório de integrais de n sub-domínios de geometrias mais simples (de volumes V_1, \dots, V_n), ou seja, o integral do volume de uma função f é obtido conforme Equação (3.3).

$$\int_V f dV = \sum_{i=1}^n \int_{V_i} f dV \quad (3.3)$$

Desta forma, em (3.3) assume-se que V segue a Equação (3.4).

$$V = \sum_{i=1}^n V_i \quad (3.4)$$

Na computação, a divisão do espaço de busca é uma estratégia geralmente aplicada para tratar ambientes multimodais. Uma aplicação desta metodologia pode ser vista em *forking Genetic Algorithm* (fGA) (TSUTSUI; FUJIMOTO; GHOSH, 1997) onde são consideradas duas opções: a primeira denominada *genotypic forking Genetic Algorithm* (g-fGA), que determina o subespaço de busca para cada subpopulação; e a segunda, denominada *phenotypic forking Genetic Algorithm* (p-fGA), que define um subespaço de busca a partir de um hiper-cubo que engloba a melhor solução atual no espaço de função. Um outro exemplo de aplicação de subdivisão do espaço de busca é o *Genetic Algorithm Search Space Splicing Particle Swarm Optimization* (GA-SSS-PSO) (LI et al., 2013). Conforme dito neste último trabalho, modificar o espaço de busca pode proporcionar uma melhor performance dos algoritmos evolucionários, todavia tem recebido pouca atenção da comunidade científica.

O objetivo em subdividir o espaço de busca é conseguir explorar melhor o ambiente, utilizando-se de subpopulações. Para verificar a viabilidade de aplicar o CMA-ES em subespaços, foram construídas algumas propostas baseadas nesse pressuposto e executados testes com as 28 funções de teste contidas na CEC-2013 (LIANG et al., 2013). A primeira proposta consiste em subdividir cada dimensão do espaço de busca original β em 2 partes iguais, sendo formados l hipercubos ($\Gamma_1, \dots, \Gamma_l$), utilizando todas as possíveis combinações entre os novos limites estipulados. Como cada dimensão é particionada ao meio, a quantidade de hipercubos pode ser obtida através da Equação (3.5):

$$l = 2^d, \quad (3.5)$$

onde d é a dimensão do espaço de busca.

A partir de então, um CMA-ES independente deve ser executado em cada subespaço, onde aquele hipercubo que ao término da execução apresentar a melhor solução deve ser escolhido para ser explorado na próxima iteração. O processo deve se repetir

de forma recursiva até que um dos critérios de parada seja alcançado. Esta abordagem apresenta uma séria limitação: se em uma determinada iteração do algoritmo o melhor indivíduo encontrado pertencer ao hipercubo Γ_i , a busca será guiada para aquele subespaço, mesmo que o ótimo global esteja em Γ_j (para $i \neq j$). Caso isso ocorra, é possível verificar que o algoritmo não mais conseguirá convergir para o ótimo global.

Com vistas à limitação apresentada, aplicou-se o método da Roleta (GOLDBERG, 1989) para selecionar qual seria o próximo hipercubo a ser explorado. Para se ter uma ideia preliminar do comportamento desta proposta, foi aplicado um teste em cada uma das funções da CEC-2013 com espaço de busca $\in \mathbb{R}^2$ para verificar se o algoritmo converge para o ótimo global. Os resultados podem ser vistos na Tabela 8, onde os valores em negrito indicam esta convergência.

Tabela 8 – Convergência com a utilização do método da Roleta para a escolha do espaço de busca (2-D)

Função Objetivo	Ótimo Global	Melhor indivíduo encontrado
1	-1400	-1400
2	-1300	-1300
3	-1200	2.56464e+08
4	-1100	-1100
5	-1000	-1000
6	-900	-900
7	-800	-771.974
8	-700	-700
9	-600	-598.209
10	-500	-473.902
11	-400	-398.01
12	-300	-248.056
13	-200	-186.84
14	-100	18.4383
15	100	100.624
16	200	200
17	300	364.49
18	400	402.54
19	500	603.498
20	600	600.156
21	700	1375.3
22	800	938.212
23	900	1713.55
24	1000	1114.8
25	1100	1208.79
26	1200	1325.87
27	1300	1410.68
28	1400	1806.13

Fonte: Próprio Autor

Pode ser observado na Tabela 8, que apesar do algoritmo atenuar as chances de desprezar a busca em um determinado hipercubo que poderia conter o ótimo global, apenas houve a convergência para o ótimo global em 7 das 28 funções testadas. Com isso, procurando prover melhorias à proposta que aplica divisão do espaço de busca, evitando que o algoritmo descarte um subespaço que pode conter a melhor solução do problema, utilizou-se o conhecimento sobre as funções objetivo que estavam sendo utilizadas para

teste, nas quais o ponto mínimo em \mathbb{R}^2 é $(-21, 11)$ para todas as funções. Sendo assim, a cada iteração do algoritmo, o espaço de busca sofria uma redução na ordem da metade dos limites superior (100) e inferior (-100). Ou seja, a opção anterior de subdivisão do espaço original β em l hipercubos, foi substituída pela compressão do espaço do espaço de busca inicial. Desta forma percebeu-se que na terceira iteração, em que os limites das dimensões do espaço de busca estavam no intervalo $[-25, 25]$, o algoritmo consegue encontrar melhores soluções, convergindo para o ótimo global em 24 das 28 funções, apresentando superioridade ao CMA-ES original (veja a Tabela 9, onde os valores em negrito indicam que o ótimo global foi encontrado). Este experimento corroborou com a suposição de que se for possível limitar o espaço de busca de forma que este consiga englobar a melhor solução global, com baixa variância em cada dimensão, o CMA-ES consegue melhores resultados.

Tabela 9 – CMA-ES versus CMA-ES com compressão do espaço de busca de $[-100,100]$ para $[-25,25]$ (2-D)

Funções Objetivo	CMA-ES (Mediana)	CMA-ES com redução do espaço de busca (Mediana)
01	-1400,0000	-1400,0000
02	-1300,0000	-1300,0000
03	-1200,0000	-1200,0000
04	-1100,0000	-1100,0000
05	-1000,0000	-1000,0000
06	-900,0000	-900,0000
07	-618,4340	-800,0000
08	-680,0000	-700,0000
09	-597,0730	-600,0000
10	-499,8740	-499,9930
11	-378,6085	-400,0000
12	-285,5735	-300,0000
13	-197,9440	-200,0000
14	350,1580	-100,0000
15	473,6480	100,0000
16	200,0000	200,0000
17	340,9780	300,0100
18	427,6280	400,0500
19	500,1230	500,0000
20	600,9675	600,0190
21	900,0000	700,0000
22	1316,6900	800,0000
23	1321,2950	900,0000
24	1196,7500	1000,0000
25	1257,5300	1100,0000
26	1356,8900	1200,0000
27	1600,0000	1300,0000
28	1700,0000	1400,0000

Fonte: Próprio Autor

A partir dos experimentos iniciais, percebe-se que explorar o espaço através de subpopulações em seus respectivos subespaços é uma proposta promissora, contudo desprezar a busca em um destes hipercubos pode fazer com que o algoritmo não convirja para o ótimo global.

Com vistas a estas análises chega-se a uma nova proposta: o espaço β inicialmente é subdividido, conforme o primeiro experimento e da mesma forma descrita, e o

CMA-ES é independentemente aplicado em cada subespaço. Esta fase é denominada de exploração. Em seguida executa-se a fase de ajuste espacial, que consiste em modificar os limites superior e inferior de cada dimensão do problema. Esta modificação se dá da seguinte forma: após o término da evolução de cada Γ_i , obtém-se o melhor indivíduo encontrado (\mathbf{x}_i); então o vetor com os limites inferiores das dimensões é obtido conforme a Equação (3.6).

$$\mathbf{limInf} = \min(x_i^z), \forall i \in l, \forall z \in d, \quad (3.6)$$

onde \mathbf{limInf}^z armazena o novo limite inferior para a z -ésima dimensão.

Analogamente, os limites superiores das dimensões são obtidos de acordo com a Equação (3.7).

$$\mathbf{limSup} = \max(x_i^z), \forall i \in l, \forall z \in d \quad (3.7)$$

O processo se repete até que um dos critérios de parada sejam alcançados. Desta forma não ocorre a eliminação de um subespaço, o que como foi visto anteriormente é ruim para o processo, contudo há uma compressão do espaço de busca, baseada nos melhores indivíduos extraídos de cada hipercubo, onde no pior caso os limites não sofrem alteração e o processo fica análogo à uma reinicialização independente. Os resultados podem ser vistos na Tabela 10.

Tabela 10 – *Mann-Whitney-Wilcoxon Test* (CMA-ES \times CMA-ES-PSS) (2-D)

Função Objetivo	CMA-ES (Mediana)	CMA-ES-PSS (Mediana)	Valor-P
01	-1400,0000000	-1400,0000000	0,3681000
02	-1300,0000000	-1300,0000000	0,0335900
03	-1200,0000000	-1200,0000000	0,0989400
04	-1100,0000000	-1100,0000000	NA
05	-1000,0000000	-1000,0000000	NA
06	5774,5250000	-900,0000000	NA
07	-687,9080000	-731,6135000	0,1613000
08	-678,8905000	-700,0000000	0,0033300
09	-585,5725000	-599,5525000	0,0201700
10	-499,9830000	-499,7435000	0,1038000
11	-384,5785000	-396,8125000	0,1027000
12	-43,3078900	-295,4345000	0,0748900
13	-160,0540000	-200,0000000	0,0028650
14	1560,9400000	121,6800000	0,0755500
15	1565,5750000	100,3120000	0,0132700
16	200,1320000	200,0000000	0,3681000
17	1092,2250000	309,3640000	0,0288100
18	1257,2550000	418,2735000	0,4359000
19	501,2695000	500,0000000	0,1838000
20	605,0000000	600,6180000	0,2258000
21	1100,1900000	700,0000000	0,0000543
22	3101,6500000	923,9800000	0,0246400
23	3546,2350000	1169,0900000	0,0583100
24	1258,5350000	1105,9650000	0,0450700
25	1358,3250000	1241,1950000	0,2723000
26	1535,6550000	1316,1800000	0,1396000
27	1864,4300000	1675,2950000	0,5650000
28	1700,0000000	1400,0000000	0,0000332

Fonte: Próprio Autor

Os resultados apresentados na Tabela 10 sugerem que utilizar o conhecimento sobre o espaço de características para restringir a busca do CMA-ES em regiões promissoras, além de realizar buscas em diferentes subespaços desta nova região, potencializa as chances do algoritmo convergir para o ótimo global.

3.5 MODELO PROPOSTO

A partir do que foi exposto no Capítulo 2 e do estudo apresentado no presente capítulo, é possível concluir que o CMA-ES possui um desempenho satisfatório quando aplicado a funções unimodais. Contudo, para ambientes multimodais ele é claramente dependente de sua inicialização, havendo algumas alternativas que podem mitigar esta limitação, dentre as quais destacam-se: ajuste do tamanho populacional e busca guiada. Na primeira aumenta-se a probabilidade de se explorar mais o ambiente no qual está inserido, mas continua sendo dependente da inicialização, também devendo ser levado em consideração que há um aumento da quantidade de avaliações de função, à medida em que há um acréscimo no tamanho populacional; já na segunda, por haver n pontos de inicialização, espera-se que a exploração seja mais uniformemente distribuída no espaço de busca, guiando o processo para regiões promissoras e reduzindo as chances de ficar preso em mínimos locais. Todavia este processo além de elevar o custo computacional, eleva de forma expressiva a quantidade de avaliações de função à medida que se aumenta n .

Partindo desta análise e baseando-se nas características previamente apresentadas, foi construído o algoritmo denominado L-CMA-DE (Algoritmo 9). Para tanto, como foi observado que a busca em diferentes regiões do espaço de características propicia uma melhor exploração do ambiente, o L-CMA-DE executa n CMA-ES em paralelo (linhas 8-12), que são inicializados a partir de n pontos aleatórios gerados através de distribuição Uniformemente neste espaço (linhas 3-5). Dado que a população de cada CMA-ES não necessita de muitos indivíduos para que seja realizada exploração, seu tamanho (λ) foi configurado conforme sugerido em Hansen (2016) (Equação (3.8)).

$$\lambda = 4 + \lfloor (3 \times \log(N)) \rfloor \quad , \quad (3.8)$$

onde N é a dimensão do espaço de busca.

Uma vez que cada CMA-ES realiza uma busca local, se faz necessário inserir diversidade, de forma a possibilitar buscas em regiões ainda não exploradas. Conforme visto no Capítulo 3, o DE, por suas características exploratórias, apresenta-se como uma alternativa promissora para provocar esta diversidade. Por este motivo decidiu-se utilizar o *DE/rand/1/bin* (Algoritmo 8), com configurações paramétricas que enfatizem a exploração, conforme visto em Epitropakis, Plagianakos e Vrahatis (2008), Coletta et al. (2015). Sendo assim, o DE é inicializado com o população (NP) de tamanho 100, sendo 10% composta pelos N indivíduos provenientes de cada CMA-ES (linha 10) e os $NP - N$ indivíduos gerados através de distribuição Uniforme. Após a execução da evolução diferencial, esta retorna os N melhores indivíduos encontrados (linha 13) e o processo se repete até que um critério de parada seja atingido (linha 14). O Algoritmo 9 descreve o arcabouço geral do modelo proposto, onde:

-
- N : é a dimensionalidade do espaço de busca;
 - *percentFES*: percentual da quantidade de avaliações de função remanescentes que cada CMA-ES deve executar ao máximo;
 - *stagCMAES*: quantidade máxima de gerações em que o CMA-ES pode ser executado sem melhorar o melhor indivíduo encontrado naquele acionamento;
 - *stagDE*: quantidade máxima de gerações em que o DE pode ser executado sem melhorar o melhor indivíduo encontrado naquele acionamento;
 - *sparsityLevel*: grau de esparsidade mínimo da matriz de covariância (\mathbf{C}) para que o CMA-ES pare;
 - *refValue*: valor de referência considerado esparso;
 - *maxGenDE*: quantidade máxima de gerações para o DE;
 - *li*: limite inferior do valor de cada dimensão do espaço de busca;
 - *ls*: limite superior do valor de cada dimensão do espaço de busca;
 - *totalContEval*: quantidade de avaliações de função consumida, ao todo;
 - *stopeval*: quantidade máxima de avaliações de função permitida, ao todo;
 - *randIndivToDE*: quantidade de indivíduos aleatórios a serem gerados cada vez que o DE é acionado;
 - \mathbf{M} : matriz com os pontos de partida, de cada CMA-ES, dispostos em colunas;
 - *valorAleatorio(a,b)*: gerador de vetor aleatório, uniformemente distribuído entre a e b ;
 - *cmes*: método que executa o CMA-ES (Algoritmo 1);
 - *de*: método que executa o DE (Algoritmo 8).

Algoritmo 8 *DE/rand/1/bin*

```

1: Entrada:  $F, Cr, NP, li, ls, d$ 
2: para  $k \leftarrow 1$  até  $NP$  faça
3:    $\mathbf{X}.coluna[k] \leftarrow \text{vetorAleatorio}(li, ls)$ 
4: fim para
5: repita
6:   para  $k \leftarrow 1$  até  $NP$  faça ▷ criar vetores mutantes
7:      $r2 \leftarrow \text{valorAleatorio}(1, NP)$ 
8:      $r3 \leftarrow \text{valorAleatorio}(1, NP)$ 
9:     enquanto  $k = r2 \parallel k = r3 \parallel r2 = r3$  faça
10:       $r2 \leftarrow \text{valorAleatorio}(1, NP)$ 
11:       $r3 \leftarrow \text{valorAleatorio}(1, NP)$ 
12:     fim enquanto
13:      $\mathbf{M}.coluna[k] \leftarrow \mathbf{X}.coluna[k] + F * (\mathbf{X}.coluna[r2] - \mathbf{X}.coluna[r3])$ 
14:   fim para
15:   para  $k \leftarrow 1$  até  $NP$  faça ▷ criar vetores tentativa (cruzamento binomial)
16:     para  $j \leftarrow 1$  até  $d$  faça
17:        $rnd \leftarrow \text{valorAleatorio}(0, 1)$ 
18:       se  $rnd \leq Cr$  então
19:          $\mathbf{T}[j, k] \leftarrow \mathbf{M}[j, k]$ 
20:       senão
21:          $\mathbf{T}[j, k] \leftarrow \mathbf{X}[j, k]$ 
22:       fim se
23:     fim para
24:     se  $\text{aptidao}(\mathbf{T}.coluna[k])$  é melhor que  $\text{aptidao}(\mathbf{X}.coluna[k])$  então
25:        $\mathbf{X}.coluna[k] \leftarrow \mathbf{T}.coluna[k]$ 
26:     fim se
27:   fim para
28: até Até que um critério de parada seja atingido

```

Algoritmo 9 Arcabouço geral do algoritmo proposto

```

1: Entrada:  $N, percentFESMAES, stagCMAES, stagDE, sparsityLevel, refValue$ 
2:    $maxGenDE, li, ls, totalCountEval, stopeval, randIndivToDE$ 
3: para  $k \leftarrow 1$  até  $N$  faça
4:    $\mathbf{M}.coluna[k] \leftarrow \text{valorAleatorio}(li, ls)$ 
5: fim para
6: repita
7:    $maxEval \leftarrow \lfloor (stopeval - totalCountEval) * percentFESMAES \rfloor$ 
8:   para  $k \leftarrow 1$  até  $N$  faça
9:      $\mathbf{x}_k \leftarrow \mathbf{M}.coluna[k]$ 
10:     $\mathbf{x}_k \leftarrow \text{cmaes}(N, maxEval, stagCMAES, \mathbf{x}_k, sparsityLevel, refValue)$ 
11:     $\mathbf{M}.coluna[k] \leftarrow \mathbf{x}_k$ 
12:   fim para
13:    $\mathbf{M} \leftarrow \text{de}(\mathbf{M}, stagDE, maxGenDE, randIndivToDE)$ 
14: até  $totalContEval = stopeval$ 

```

Para o algoritmo desenvolvido neste trabalho, ao ser executada a Linha 10

do Algoritmo 9, o Algoritmo 1 é acionado, com a modificação dos critérios de análise de convergência (Linha 9), que passam a ser: *maxEval*, *stagCMAES* e *sparsityLevel*. Estes critérios são necessários, uma vez que, se a região que estiver sendo explorada for unimodal, ou multimodal com formatos diferentes, o grau de esparsidade³ da matriz de covariância (*sparsityLevel*) tende a aumentar (ver Tabela 1), indicando que os indivíduos são muito semelhantes entre si, muito embora suas aptidões sejam muito diferentes; se a região é um platô, os indivíduos serão diferentes e o grau de esparsidade da matriz de covariância será baixo, contudo suas aptidões serão semelhantes, tornando difícil melhorá-las (o que será identificado com o critério *stagCMAES*); ainda assim, podem haver funções híbridas que não atinjam nenhum dos dois critérios citados, e como o objetivo é que o CMA-ES avance um pouco a cada iteração, é estabelecido uma quantidade máxima de avaliação de funções (*maxEval*). Quantidade esta que é estabelecida a partir de um percentual da quantidade de avaliações de função restantes (*percentFESCMAES*), de forma a permitir que cada CMA-ES independente seja executado.

A execução do DE (Linha 13 do Algoritmo 9) é realizada conforme Algoritmo 8, cuja população inicial é de tamanho 100, devido à estudos que demonstram que esse tamanho populacional propicia uma maior diferença entre os indivíduos, logo diversidade, como pode ser visto em Piotrowski (2017). Esta população é formada arbitrariamente com 10% dos indivíduos provenientes dos CMA-ES (ou todos eles, caso a quantidade seja menor), para que não se perca o conhecimento sobre o ambiente, sendo os demais gerados aleatoriamente, uniformemente distribuídos no espaço de busca, para que se aumente as chances de explorar distintas regiões. Já os valores de F e Cr , foram inseridos como 0,7 e 1, respectivamente. O valor de Cr foi assim selecionado para que o vetor tentativa (Linha 19 do Algoritmo 8) sempre seja o vetor mutante, provocando assim maior exploração do espaço de busca. Ainda nesse sentido o valor de F foi inserido, conforme estudo realizado em Zielinski et al. (2006). Os critérios de parada do DE foram selecionados de forma que fosse possível pará-lo, caso não seja observada uma melhoria (*stagDE*), e que as buscas sejam limitadas (*maxGenDE*), para que os CMA-ES tenham maiores chances de serem acionados novamente.

³ Para acelerar a indicação de elevado grau de esparsidade, foi escolhido um valor de referência próximo de zero (*refValue*). Logo, o grau de esparsidade da matriz de covariância contabilizará o percentual de entradas que possuem este valor

4 VALIDAÇÃO DO MODELO

Neste Capítulo serão apresentados os testes (simulações) com o algoritmo proposto para sua validação. Estes experimentos incluem estudos quanto à sensibilidade (Seção 4.2) e ajuste paramétricos (Seção 4.3), bem a escolha dos algoritmos para comparação de resultados (Seção 4.5). São também tratadas funções (Seção 4.4), métricas (Seção 4.6) e testes estatísticos (Seção 4.7) utilizados para a validação da proposta, mediante os resultados apresentados (Seção 4.8).

Os testes apresentados neste Capítulo tem por objetivo validar o L-CMA-DE, analisando a qualidade das soluções obtidas por ele, com foco nas funções objetivo multimodais, comparando-as com o CMA-ES e com algoritmos evolucionários Estado-da-Arte que sejam baseados neste.

4.1 COMPLEXIDADE DE TEMPO

Conforme indicado em Hansen, Müller e Koumoutsakos (2003), Auger e Hansen (2012), a complexidade de tempo do modelo CMA-ES é $\mathcal{O}(N^2)$, sendo N as dimensões do espaço de busca. A complexidade de tempo do algoritmo proposto (L-CMA-DE) pode ser obtida a partir da seguinte análise: no Algoritmo 9, o custo das linha 3 a 5 é $\mathcal{O}(N)$. A instrução "**para**", da linha 8 à 12, possui o custo $\mathcal{O}(N)$ multiplicado pelo custo do CMA-ES (linha 10), logo, o custo desta instrução é igual a $\mathcal{O}(N \times N^2) = \mathcal{O}(N^3)$. O custo da execução do DE (linha 13), deve ser analisado a partir do Algoritmo 8, onde da linha 2 a 4 o custo é $\mathcal{O}(NP)$, sendo NP o tamanho da população do DE. Como o laço iniciado na linha 5, no pior caso, é executado $maxGenDE$ vezes, o custo de realizar as instruções dentro dele deve ser multiplicado por $\mathcal{O}(maxGenDE)$. Utilizando amostragem sem reposição, a criação de vetores mutantes (linhas 6 a 14) tem custo $\mathcal{O}(NP)$. Sendo $d = N$, a criação de vetores tentativa (linhas 15 a 27) possui custo $\mathcal{O}(NP \times N)$. Sendo assim, o custo da execução do DE é igual a $\mathcal{O}(NP) + \mathcal{O}(maxGenDE) \times \mathcal{O}(NP \times N) = \mathcal{O}(maxGenDE \times NP \times N)$. Logo, pode-se afirmar que o custo do modelo L-CMA-DE é igual a $\mathcal{O}(N^3) + \mathcal{O}(maxGenDE \times NP \times N) = \mathcal{O}(N^3)$

4.2 SENSIBILIDADE PARAMÉTRICA

Para realizar a análise de sensibilidade e ajuste paramétricos do algoritmo proposto, foi utilizado o método Amostragem de Hipercubo Latino, do inglês *Latin Hypercube Sampling* (LHS). Proposto inicialmente em McKay, Beckman e Conover (1979), o LHS gera uma amostra de tamanho n a partir de k variáveis X_1, \dots, X_k . Os limites de cada variável são divididos em n intervalos, sem sobreposição. Um valor de cada intervalo é selecionado aleatoriamente, com igual probabilidade de seleção. Os n valores obtidos para

X_1 são emparelhados de uma maneira aleatória com os n valores de X_2 . Esses n pares são combinados de maneira aleatória com os n valores de X_3 , para formar n conjuntos com 3 valores, e assim por diante, até que um conjunto de n k -tuplas seja formado. Este conjunto de k -tuplas é o espaço amostral do hipercubo latino. Desta forma, existem $(n!)^{k-1}$ possíveis combinações de intervalos para uma amostragem de hipercubo latino.

Iman e Helton (1988) compara a utilização de algumas técnicas de análise de sensibilidade paramétrica e constata a boa performance em utilizar regressão aplicada ao LHS. Portanto, foi realizada a análise de sensibilidade dos parâmetros do modelo proposto utilizando regressão linear, utilizando o método *LinearRegression* da biblioteca *Scikit-learn* em Python (PEDREGOSA et al., 2011). Foram empregados 500 conjuntos paramétricos gerados com o LHS e as aptidões obtidas com cada um destes conjuntos. A partir desta análise é realizada a escolha dos valores dos parâmetros. Inicialmente arbitraram-se os limites dos parâmetros, de forma a representar baixos e elevados valores paramétricos, conforme abaixo:

- $percentFESCMAES = [0,01 \dots 0,1]$;
- $stagCMAES = \{20, \dots, 100\}$;
- $stagDE = \{10, \dots, 100\}$;
- $sparsityLevel = [0,05 \dots 0,9]$;
- $refValue = [0,0005 \dots 0,005]$;
- $maxGenDE = \{10, \dots, 100\}$.

Para realização da análise de sensibilidade paramétrica foram utilizadas as funções da CEC-2017 (AWAD et al., 2016). A análise pode ser realizada observando-se as Figuras 10 - 19, onde os eixos x e y representam os valores paramétricos e as aptidões, respectivamente. Nestas figuras, os pontos azuis representam as aptidões ao final de cada execução e a linha em vermelho é obtida através de Regressão Linear, para que seja possível analisar qual o intervalo se mostra mais promissor. Assim, como as funções utilizadas são de minimização, quanto menor o valor no eixo y melhor é a solução. Para facilitar o entendimento sobre as variações dos valores paramétricos, as funções foram separadas em três classes, conforme o comportamento apresentado:

- Mínimo: indica que quanto menor o valor do parâmetro em estudo, melhor a aptidão obtida;
- Neutro: indica que a aptidão obtida é pouco sensível ao parâmetro em estudo, já que qualquer valor dentro do intervalo estabelecido tem elevada chance de produzir indivíduos finais com aptidões semelhantes. Logo, a reta em vermelho deve ficar aproximadamente na horizontal;

- Máximo: indica que quanto maior o valor do parâmetro, melhor a aptidão resultante.

4.2.1 Percentual do Total de Avaliações de Função Remanescentes

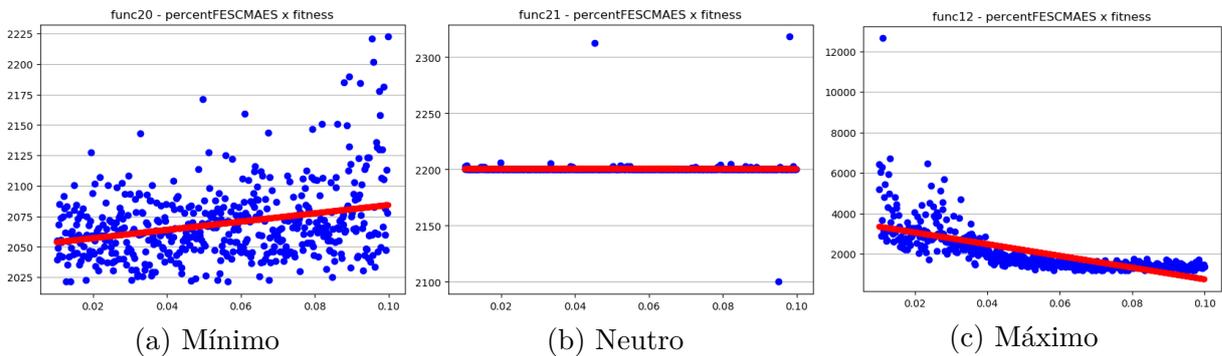
A Tabela 11 expõe as classificações das funções em relação à variação do parâmetro *percentFESMAES*. A Figura 10a ilustra a classe Mínimo observada na função 20. A Figura 10b ilustra a classe Neutra observada na função 21. Já a Figura 10c ilustra a classe Máximo identificada na função 12.

Tabela 11 – Comportamento das funções quanto à variação de *percentFESMAES*

Classificação	Funções
Mínimo	9, 10, 16, 17, 20, 25, 29
Neutro	5, 6, 7, 8, 11, 21, 22, 23, 24, 26, 27, 28
Máximo	1, 3, 4, 12, 13, 14, 15, 18, 19, 30

Fonte: Próprio Autor

Figura 10 – Representação da sensibilidade ao parâmetro *percentFESMAES*



Fonte: Próprio Autor

É possível perceber que na maioria das funções o L-CMA-DE é pouco sensível aos valores de *percentFESMAES* e que as funções que possuem o comportamento Mínimo, apresentam alto valor de dispersão. Todavia as funções que apresentam o comportamento Máximo possuem dispersão menor na região em que o valor possibilita a otimização da aptidão.

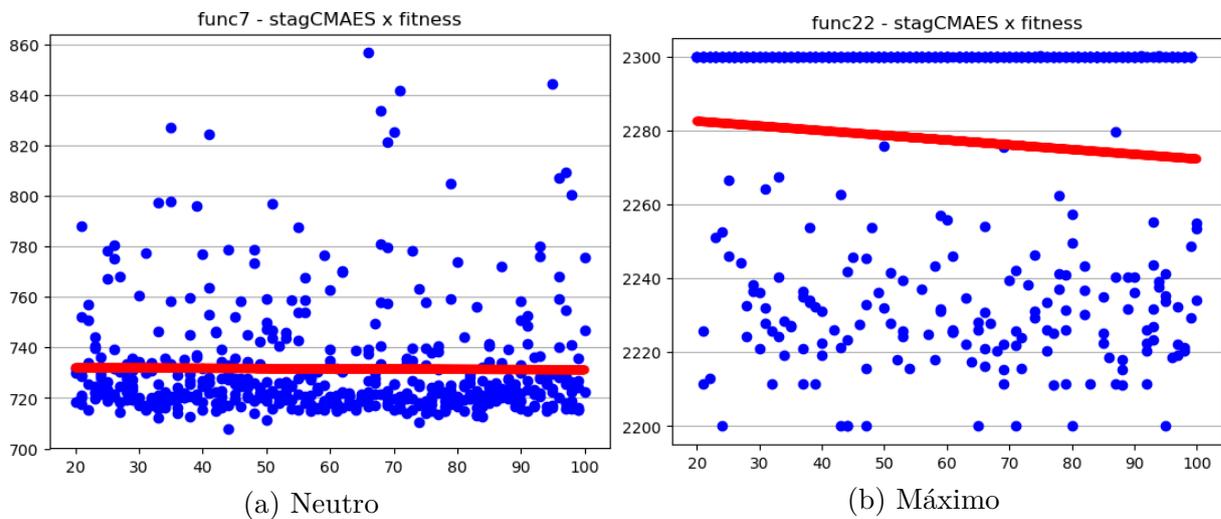
4.2.2 Quantidade de Gerações do CMA-ES sem Melhorar o Melhor Indivíduo

A Tabela 12 apresenta as classificações das funções em relação à variação do parâmetro. Não foi observada nenhuma função que apresentasse o comportamento Mínimo. A Figura 11a ilustra a classe Neutra observada na função 7. Já a Figura 11b ilustra a classe Máximo identificada na função 22.

Tabela 12 – Comportamento das funções quanto à variação de *stagCMAES*

Classificação	Funções
Mínimo	
Neutro	1, 3, 4, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 27, 28, 29, 30
Máximo	5, 6, 9, 22, 26

Fonte: Próprio Autor

Figura 11 – Representação da sensibilidade do parâmetro *stagCMAES*

Fonte: Próprio Autor.

Observa-se na Figura 11 que o algoritmo não é sensível ao parâmetro *stagCMAES*.

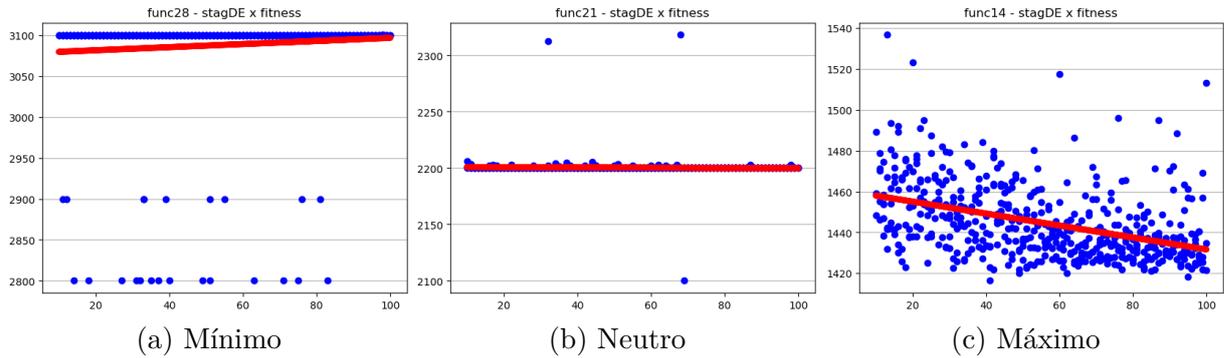
4.2.3 Quantidade de Gerações do DE sem Melhorar o Melhor Indivíduo

A Tabela 13 expõe as classificações das funções em relação à variação do parâmetro *stagDE*. A Figura 12a ilustra a classe Neutra observada na função 28. A Figura 12b ilustra a classe Neutra observada na função 21. Já a Figura 12c ilustra a classe Máximo identificada na função 14.

Tabela 13 – Comportamento das funções quanto à variação de *stagDE*

Classificação	Funções
Mínimo	26, 28
Neutro	1, 3, 4, 7, 10, 11, 12, 13, 17, 18, 19, 21, 23, 25, 27, 30
Máximo	5, 6, 8, 9, 14, 15, 16, 20, 22, 24, 29

Fonte: Próprio Autor

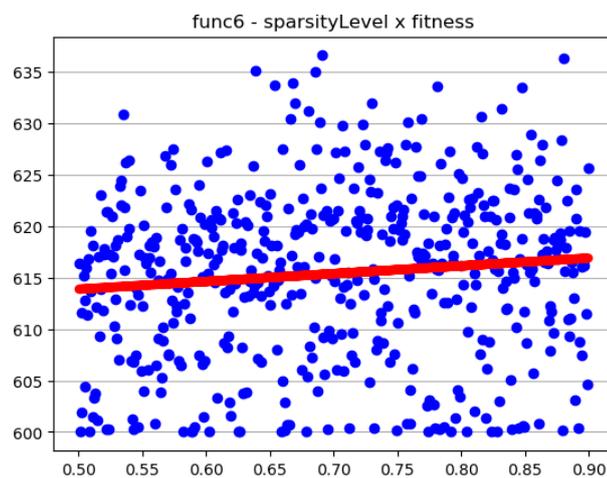
Figura 12 – Representação da sensibilidade ao parâmetro *stagDE*

Fonte: Próprio Autor

É possível perceber que em algumas funções o L-CMA-DE apresenta uma sensibilidade a este parâmetro, já naquelas em que não se observa esta sensibilidade há um comportamento bastante estável ao intervalo numérico estipulado.

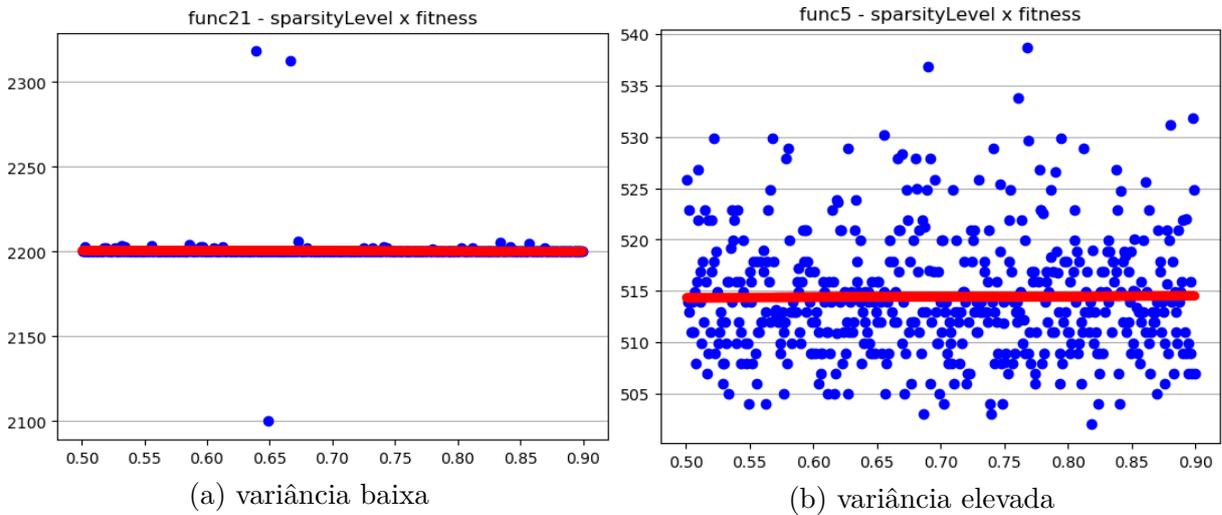
4.2.4 Grau de Esparsidade da Matriz de Covariância

Para o parâmetro *sparsityLevel*, com exceção da função 6 (Figura 13) que apresentou o comportamento Mínimo, todas as outras apresentaram um comportamento neutro, apenas com diferenças na variância dos dados, como pode ser visto na Figura 14.

Figura 13 – Representação da sensibilidade ao parâmetro *sparsityLevel* (comportamento Mínimo)

Fonte: Próprio Autor

Figura 14 – Representação da sensibilidade ao parâmetro *sparsityLevel* (comportamento Neutro)

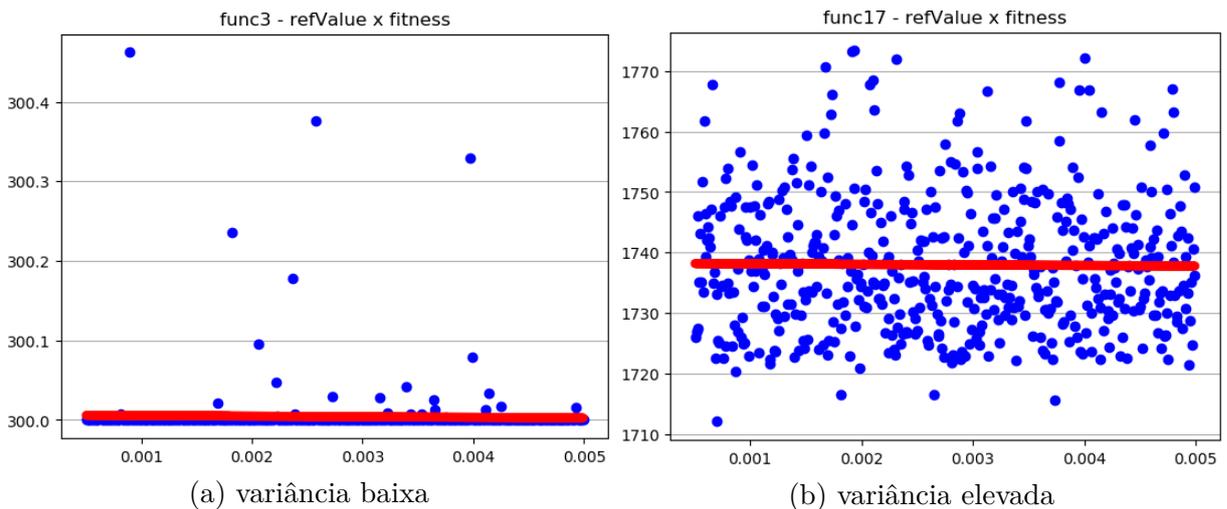


Fonte: Próprio Autor

4.2.5 Valor de Referência Considerado como Esparso

O comportamento observado em relação ao parâmetro *refValue* é semelhante àquele apresentado quanto ao parâmetro *sparsityLevel* (Seção 4.2.4), apresentando apenas o comportamento Neutro, apenas com diferença nos valores da variância (Figura 15).

Figura 15 – Representação da sensibilidade ao parâmetro *refValue* (comportamento Neutro)



Fonte: Próprio Autor

4.2.6 Quantidade Máxima de Gerações para o DE

A Tabela 14 apresenta as classificações das funções em relação à variação do parâmetro *maxGenDE*. Apenas as funções 25 e 26 apresentaram o comportamento Mínimo (Figura 16a), sem um comportamento estável. A Figura 16b ilustra a classe

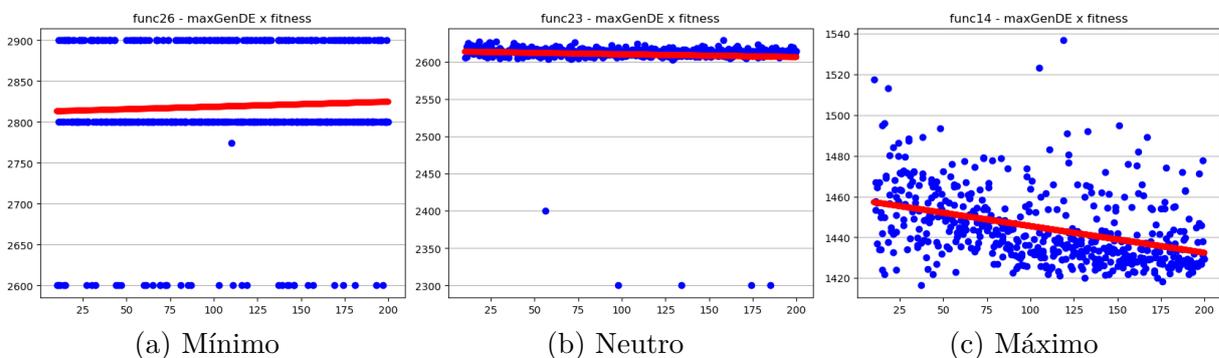
Neutro observada na função 23. Já a Figura 16c ilustra a classe Máximo identificada na função 14.

Tabela 14 – Comportamento das funções quanto à variação de $maxGenDE$

Classificação	Funções
Mínimo	25, 26
Neutro	1, 3, 4, 10, 11, 12, 13, 18, 20, 21, 22, 23, 24, 27, 29
Máximo	5, 6, 7, 8, 9, 14, 15, 16, 17, 19, 28, 30

Fonte: Próprio Autor

Figura 16 – Representação da sensibilidade ao parâmetro $maxGenDE$



Fonte: Próprio Autor

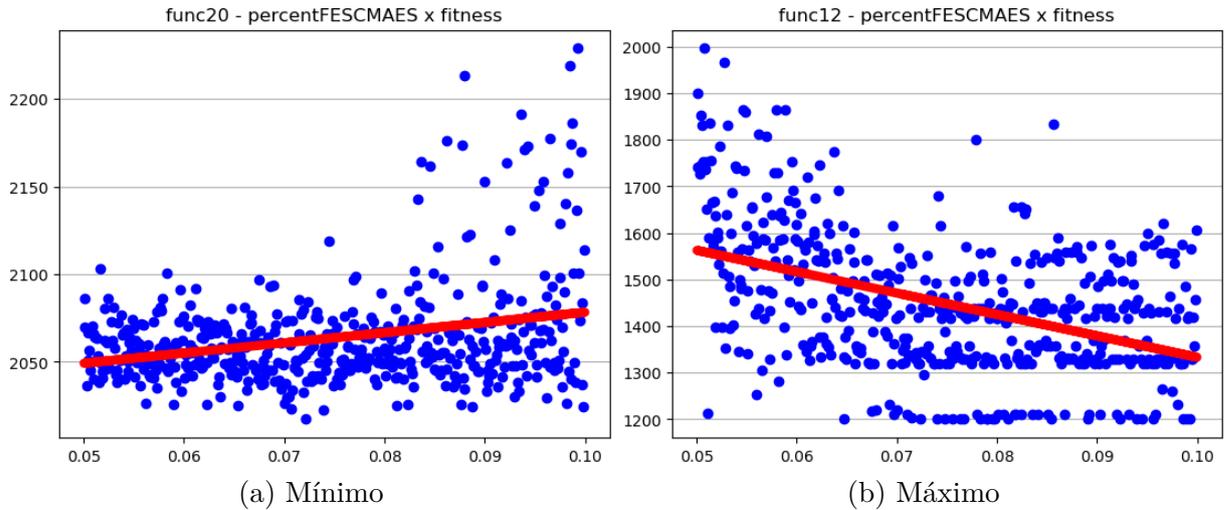
4.3 AJUSTE PARAMÉTRICO

Após a análise de sensibilidade realizada na Seção 4.2, levando em consideração a dispersão dos valores obtidos e a complexidade das funções (optando-se em provocar melhores resultados nas funções mais complexas), foram estabelecidos novos intervalos para executar o LHS, a fim de uma melhor escolha dos valores paramétricos. Os parâmetros $refValue$ e $maxGenDE$ foram fixados em 0,0035 e 190, respectivamente, devido ao comportamento observado. Segue novos intervalos:

- $percentFESCMAES = [0,05 \dots 0,1]$;
- $stagCMAES = \{80, \dots, 100\}$;
- $stagDE = \{80, \dots, 100\}$;
- $sparsityLevel = [0,4 \dots 0,6]$.

Para o parâmetro $percentFESCMAES$ o intervalo entre $[0,07$ e $0,08]$ apresentam-se como uma boa escolha, já que satisfazem os comportamento Mínimo e Máximo observados, como pode ser visto na Figura 17.

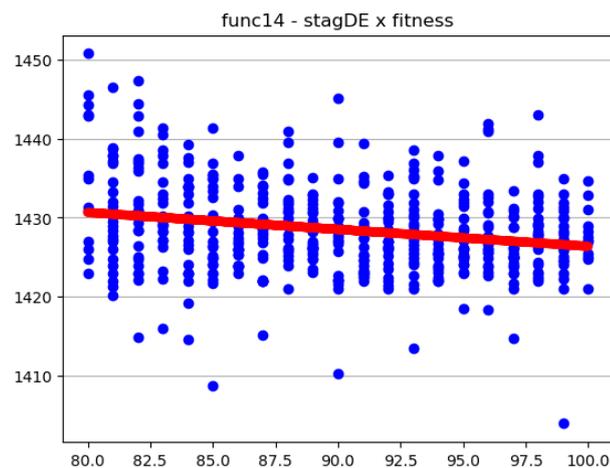
Figura 17 – Comportamento quanto à variação do parâmetro *percentFESMAES*



Fonte: Próprio Autor

Quanto ao parâmetro *stagCMAES*, como há uma grande variabilidade dos resultados, existindo entretanto um indicativo de que valores elevados deste parâmetro possibilite um melhor desempenho do algoritmo, deve-se escolher um valor que satisfaça esta análise. O comportamento observado a partir dos valores do *stagDE*, muito embora indique que uma escolha de valores elevados seja um bom caminho, não foi observado um comportamento linear (Figura 18), portanto valores mais centrais se apresentam como uma boa escolha.

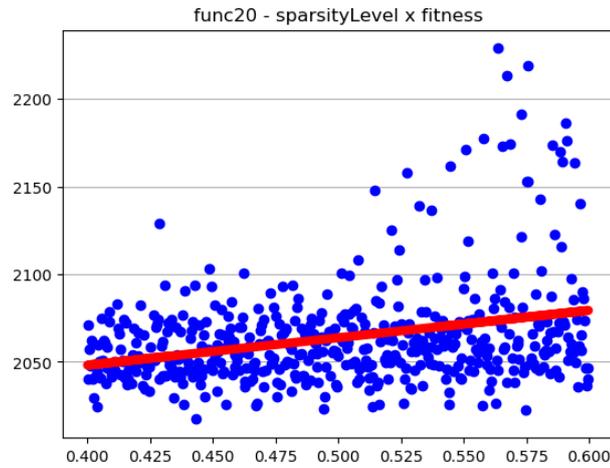
Figura 18 – Comportamento quanto à variação do parâmetro *stagDE*



Fonte: Próprio Autor

Por fim, no que diz respeito ao parâmetro *sparsityLevel*, para o segundo intervalo numérico estabelecido, além da função 6, a função 20 também apresentou o comportamento Mínimo (Figura 19). Como para as demais funções o comportamento observado foi Neutro, então deve-se optar por valores menores.

Figura 19 – Comportamento quanto à variação do parâmetro *sparsityLevel*



Fonte: Próprio Autor

Descartando-se o comportamento Neutro, uma vez que qualquer um dos valores dentro do intervalo satisfazem o objetivo, e buscando solucionar as funções mais difíceis, os valores paramétricos após o ajuste foram determinados conforme Tabela 15.

Tabela 15 – Valores paramétricos utilizados para funções da CEC-2017

Parâmetro	Valor
<i>percentFESMAES</i>	0,075
<i>stagCMAES</i>	94
<i>stagDE</i>	86
<i>sparsityLevel</i>	0,43
<i>refValue</i>	0,0035
<i>maxGenDE</i>	190

Fonte: Próprio Autor

O mesmo procedimento descrito nesta Seção e na Seção 4.2, aplicados às funções da CEC-2017 (AWAD et al., 2016), foram realizados às funções da CEC-2013 (LIANG et al., 2013), chegando-se aos valores paramétricos da Tabela 16.

Tabela 16 – Valores paramétricos utilizados para funções da CEC-2013

Parâmetro	Valor
<i>percentFESCMAES</i>	0,035
<i>stagCMAES</i>	53
<i>stagDE</i>	90
<i>sparsityLevel</i>	0,77
<i>refValue</i>	0,0048
<i>maxGenDE</i>	190

Fonte: Próprio Autor

4.4 FUNÇÕES PARA TESTES

Os testes serão executados com as funções para testes da Seção Especial da CEC-2013 (LIANG et al., 2013) e 2017 para competição em otimização de funções de um único objetivo, com espaço de busca no domínio dos números reais, descritas em Liang et al. (2013) e Awad et al. (2016), respectivamente. Todas as funções são de minimização, sem restrição, com o espaço de busca $\beta \in \{-100, \dots, 100\}^D$, no qual D representa a dimensionalidade do espaço de características. As dimensões de todas as funções são escaláveis. As características básicas das funções podem ser vistas nas Tabelas 17 e 18, onde $F_f(\mathbf{x}^*)$ representa o mínimo global da função f .

Tabela 17 – Funções da CEC-2013

Classe	f	Nome da Função	$F_f(\mathbf{x}^*)$
Unimodais	1	Sphere Function	-1400
	2	Rotated High Conditioned Elliptic Function	-1300
	3	Rotated Bent Cigar Function	-1200
	4	Rotated Discus Function	-1100
	5	Different Powers Function	-1000
Multimodais Básicas	6	Rotated Rosenbrock's Function	-900
	7	Rotated Schaffers F7 Function	-800
	8	Rotated Ackley's Function	-700
	9	Rotated Weierstrass Function	-600
	10	Rotated Griewank's Function	-500
	11	Rastrigin's Function	-400
	12	Rotated Rastrigin's Function	-300
	13	Non-Continuous Rotated Rastrigin's Function	-200
	14	Schwefel's Function	-100
	15	Rotated Schwefel's Function	100
	16	Rotated Katsuura Function	200
	17	Lunacek Bi_Rastrigin Function	300
	18	Rotated Lunacek Bi_Rastrigin Function	400
Funções Compostas	19	Expanded Griewank's plus Rosenbrock's Function	500
	20	Expanded Scaffer's F6 Function	600
	21	Composition Function 1 (n=5,Rotated)	700
	22	Composition Function 2 (n=3,Unrotated)	800
	23	Composition Function 3 (n=3,Rotated)	900
	24	Composition Function 4 (n=3,Rotated)	1000
	25	Composition Function 5 (n=3,Rotated)	1100
	26	Composition Function 6 (n=5,Rotated)	1200
27	Composition Function 7 (n=5,Rotated)	1300	
28	Composition Function 8 (n=5,Rotated)	1400	

Fonte: Adaptado de Liang et al. (2013)

Tabela 18 – Funções da CEC-2017

Classe	f	Nome da Função	$F_f(\mathbf{x}^*)$
Unimodais	1	<i>Shifted and Rotated Bent Cigar Function</i>	100
	2	<i>Shifted and Rotated Sum of Different Power Function</i>	200
	3	<i>Shifted and Rotated Zakharov Function</i>	300
Multimodais Simples	4	<i>Shifted and Rotated Rosenbrock's Function</i>	400
	5	<i>Shifted and Rotated Rastrigin's Function</i>	500
	6	<i>Shifted and Rotated Expanded Scaffer's F6 Function</i>	600
	7	<i>Shifted and Rotated Lunacek Bi_Rastrigin Function</i>	700
	8	<i>Shifted and Rotated Non-Continuous Rastrigin's Function</i>	800
	9	<i>Shifted and Rotated Levy Function</i>	900
	10	<i>Shifted and Rotated Schwefel's Function</i>	1000
Híbridas	11	<i>Hybrid Function 1 (N=3)</i>	1100
	12	<i>Hybrid Function 2 (N=3)</i>	1200
	13	<i>Hybrid Function 3 (N=3)</i>	1300
	14	<i>Hybrid Function 4 (N=4)</i>	1400
	15	<i>Hybrid Function 5 (N=4)</i>	1500
	16	<i>Hybrid Function 6 (N=4)</i>	1600
	17	<i>Hybrid Function 6 (N=5)</i>	1700
	18	<i>Hybrid Function 6 (N=5)</i>	1800
	19	<i>Hybrid Function 6 (N=5)</i>	1900
	20	<i>Hybrid Function 6 (N=6)</i>	2000
Compostas	21	<i>Composition Function 1 (N=3)</i>	2100
	22	<i>Composition Function 2 (N=3)</i>	2200
	23	<i>Composition Function 3 (N=4)</i>	2300
	24	<i>Composition Function 4 (N=4)</i>	2400
	25	<i>Composition Function 5 (N=5)</i>	2500
	26	<i>Composition Function 6 (N=5)</i>	2600
	27	<i>Composition Function 7 (N=6)</i>	2700
	28	<i>Composition Function 8 (N=6)</i>	2800
	29	<i>Composition Function 9 (N=3)</i>	2900
	30	<i>Composition Function 10 (N=3)</i>	3000

Fonte: Adaptado de Awad et al. (2016)

Nas Tabelas 17 e 18, as funções da classe Unimodal são caracterizadas pelo mínimo global bem definido, conforme pode ser visto na Figura 20a¹. A classe de funções Multimodais é caracterizada por funções que possuem diversos mínimos locais (Figura 20b)¹. Já as funções compostas são obtidas através da Equação (4.1) (ver Figura 20c)¹.

$$F(\mathbf{x}) = \sum_{i=1}^N \{\omega_i \times [\lambda_i g_i(\mathbf{x}) + bias_i]\} + F^* \quad , \quad (4.1)$$

¹ Os eixos $\in [-100, 100]$ são as dimensões do espaço de busca e o eixo vertical a aptidão

onde N é o número de funções utilizadas para construção da função composta $F(\mathbf{x})$; $g_i(\mathbf{x})$ é a i -ésima função utilizada; $bias_i$ define qual ótimo é ótimo global da i -ésima função; λ_i é utilizado para controlar $g_i(\mathbf{x})$; F^* representa o ótimo global de $F(\mathbf{x})$; e ω_i é o peso normalizado associado a i -ésima função, o qual é computado conforme a Equação (4.2).

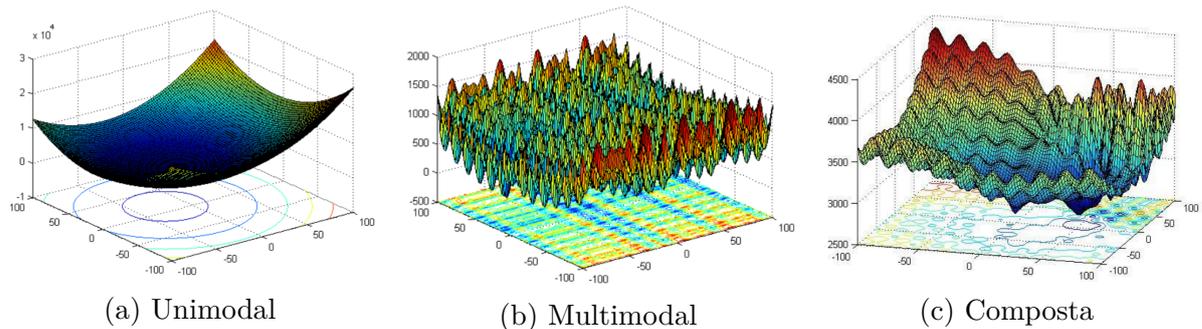
$$\omega_i = \frac{w_i}{\sum_{i=1}^N w_i} \quad , \quad (4.2)$$

sendo w_i obtido através da Equação (4.3).

$$w_i = \frac{1}{\sqrt{\sum_{j=1}^D (x_j - o_{ij})^2}} \exp\left(-\frac{\sum_{j=1}^D (x_j - o_{ij})^2}{2D\sigma_i^2}\right) \quad , \quad (4.3)$$

onde o_i é a nova posição ideal deslocada para $g_i(\mathbf{x})$; D é a dimensão do espaço de busca; e σ_i é usado para controlar o alcance de cobertura de $g_i(\mathbf{x})$.

Figura 20 – Representação em 3-D dos tipos de funções (espaço de busca em 2-D)



Fonte: Liang et al. (2013) e Awad et al. (2016)

Na Tabela 18 também existe funções Híbridas, as quais são formuladas de acordo com a Equação (4.4).

$$F(\mathbf{x}) = \left[\sum_{i=1}^N g_i(\mathbf{M}_i \mathbf{z}_i) \right] + F^* \quad , \quad (4.4)$$

sendo:

- N é o número de funções utilizadas para construção da função híbrida $F(\mathbf{x})$; $g_i(\mathbf{x})$ é a i -ésima função utilizada;
- \mathbf{M}_i é a matriz de rotação aplicada à i -ésima função;
- $\mathbf{z}_1 = [\mathbf{y}_{S_1}, \mathbf{y}_{S_2}, \dots, \mathbf{y}_{S_{n_1}}]$, $\mathbf{z}_2 = [\mathbf{y}_{S_{n_1+1}}, \mathbf{y}_{S_{n_1+2}}, \dots, \mathbf{y}_{S_{n_1+n_2}}]$, \dots ,
 $\mathbf{z}_N = [\mathbf{y}_{s_{\sum_{i=1}^{N-1} n_{i+1}}}, \mathbf{y}_{s_{\sum_{i=1}^{N-1} n_{i+2}}}, \dots, \mathbf{y}_{s_D}]$;
- $\mathbf{y} = \mathbf{x} - \mathbf{o}_i$;
- $S = rand(1 : D)$, sendo D a dimensão do espaço de busca;

- n_i é a dimensão para a i -ésima função, de modo que $\sum_{i=1}^N n_i = D$;
- $n_i = \lceil p_i D \rceil$;
- p_i é utilizado para controlar o percentual de $g_i(\mathbf{x})$.

Algumas funções compostas da CEC-2017 utilizam funções híbridas, o que dificulta ainda mais o processo de otimização.

4.5 ALGORITMOS PARA COMPARAÇÃO

Foi escolhido um conjunto de algoritmos para comparações buscando situar o desempenho do método proposto entre os modelos de interesse na família CMA-ES. As comparações envolvem o modelo original, modelos que tiveram resultados importantes em duas competições do IEEE CEC, em 2013 e 2017, que utilizaram as funções descritas na Seção 4.4 e modelos que devem lidar com funções multimodais. Inicialmente, os testes visaram mostrar que os algoritmos utilizados para a construção do modelo proposto (CMA-ES e DE), quando aplicados isoladamente, possuem comportamento muito dependente da função a ser otimizada. Neste caso, foram realizados testes comparando-se a performance de ambos em relação ao L-CMA-DE, utilizando as funções descritas na Seção 4.4.

Em seguida, foram realizados testes com o CMA-ES a fim de comprovar que seu desempenho é alto em funções unimodais, contudo, perde desempenho de modo significativo quando é empregado em funções multimodais. Logo, espera-se que este apresente uma boa performance nas primeiras funções de testes e que esta decaia à medida que seja testado em funções de complexidade maior.

Os testes realizados com o DE procuram comprovar que este consegue resultados competitivos quando aplicado em funções multimodais, devido suas características exploratórias, contudo não possibilitando uma exploração eficaz em regiões promissoras. Logo, espera-se que o algoritmo proposto apresente melhor desempenho frente a este algoritmo.

Como o procedimento do algoritmo proposto de acionar o DE e retornar ao CMA-ES pode ser entendido como uma reinicialização preservando conhecimento anterior do espaço de busca, foram realizadas comparações entre o IPOPOP-CMA-ES (AUGER; HANSEN, 2005), que é o Estado-da-Arte nesta modalidade, o qual aumenta o tamanho da população a cada reinicialização. Também serão realizados testes com duas de suas variações, que são: o NBIPOP-aCMA-ES (LOSHCHILOV, 2013), que utiliza uma estratégia aumento e/ou redução do tamanho da população; e o RB-IPOP-CMAES (BIEDRZYCKI, 2017) que utiliza o ponto médio da população com uma aproximação do ponto ótimo para a reinicialização.

Como no algoritmo proposto, o CMA-ES também pode ser considerado como um método de *super-fit*² para o DE, foram realizados testes com o CMA-ES-RIS (CARAFFINI et al., 2013), que realiza uma busca guiada, através da técnica de *super-fit*. Neste algoritmo, o CMA-ES é o responsável por esta fase, seguido de dois outros mecanismos, responsáveis por exploração e exploração, que são acionados em *loop* até o término da execução. Também comparou-se o algoritmo proposto com o LSHADE-SPACMA (MOHAMED et al., 2017), que utiliza o CMA-ES e o DE em paralelo, aumentando a probabilidade de um indivíduo ser gerado por aquele algoritmo que possui um melhor histórico de geração de prole.

4.6 MÉTRICAS

As métricas utilizadas para avaliar a qualidade das soluções geradas por cada algoritmo, para cada função utilizada, são analisadas em relação ao erro, que é caracterizado pela diferença entre a aptidão encontrada e a aptidão do ótimo global da função a ser otimizada. Sejam elas:

- mediana dos erros;
- erro médio;
- desvio padrão dos erros;
- menor erro.

O erro médio da função f é calculado conforme Equação (4.5).

$$\bar{e}_f = \frac{1}{T} \sum_{t=1}^T (F_f(\mathbf{b}_t) - F_f(\mathbf{x}^*)) \quad , \quad (4.5)$$

onde T é a quantidade de testes executados; $F_f(\mathbf{b}_t)$ e $F_f(\mathbf{x}^*)$ são os valores das aptidões do melhor indivíduo encontrado no teste t e do ótimo global, respectivamente.

O desvio padrão dos erro referente à uma função f é obtido através da Equação (4.6).

$$\sigma_f = \sqrt{\frac{\sum_{t=1}^T (F_f(\mathbf{b}_t) - \bar{e}_f)^2}{T - 1}} \quad , \quad (4.6)$$

o denominador dentro do radical é igual a $T - 1$ por se tratar de uma amostra.

² A técnica de *super-fit* consiste em inserir uma solução com elevada aptidão no início de um processo de otimização, obtida a partir de um outro algoritmo (IACCA et al., 2011)

4.7 CONFIGURAÇÃO EXPERIMENTAL

Os algoritmos CMA-ES, DE e aqueles desenvolvidos durante a pesquisa, incluindo o L-CMA-DE, foram implementados em linguagem *C++* e validados, ao passo que os resultados obtidos pelos demais modelos foram obtidos a partir da disponibilização destes por parte das competições (CEC-2013 e CEC-2017). A máquina em que foram executados os algoritmos implementados e os testes estatísticos possuía 16GB de memória RAM e processador Intel Core i7-7700 de 3,60GHz, com 4 núcleos e 8 *Threads*.

Para todos os algoritmos testados, foram executados 51 testes independentes com as funções descritas na Seção 4.4, número de execuções exigidos nas competições do IEEE CEC de 2013 e 2017, com objetivo de extrair os valores para cômputo das métricas abordadas na Seção 4.6. Os valores dos parâmetros do L-CMA-DE foram ajustados e estão informados nas Tabelas 15 e 16, ao passo que os valores paramétricos dos algoritmos comparados foram escolhidos conforme indicação de seus autores.

Como não é trivial determinar de qual distribuição provêm os resultados obtidos com as métricas descritas na Seção 4.6, foi realizada a escolha de um teste estatístico não-paramétrico. Uma vez que as amostras são independentes e as comparações serão realizadas par a par, para que seja possível uma análise mais precisa do desempenho do algoritmo proposto, optou-se por utilizar o teste de Wilcoxon-Mann-Whitney (AHMAD, 1996), que é um teste estatístico não-paramétrico para 2 amostras independentes. Trata-se de um teste para mediana, que busca determinar se duas amostras provêm da mesma população. Desta forma, assumindo-se que $Med(P)$ seja a mediana de uma população P , dadas duas populações (P_1 e P_2), a hipótese nula deste teste pode ser descrita conforme Equação (4.7).

$$H_0 : Med(P_1) = Med(P_2) \quad (4.7)$$

Sendo assim, em todos os testes realizados, foi utilizado um nível de significância de 5% ($\alpha = 0,05$).

4.8 RESULTADOS E ANÁLISES

Nesta Seção estão apresentados e discutidos os resultados dos testes estatísticos (Seção 4.7) realizados com os algoritmos mencionados na Seção 4.5. Em todas tabelas de resultados desta Seção, as médias estatisticamente melhores estão apresentadas em **negrito**. As Seções 4.8.1 e 4.8.2 apresentam os testes com as funções da CEC de 2013 e 2017, respectivamente.

4.8.1 Testes com as funções da CEC-2013

A Tabela 19 apresenta os resultados dos testes comparativos entre o L-CMA-DE e o CMA-ES, evidenciando a superioridade do algoritmo proposto frente ao CMA-ES. Os resultados apresentam superioridade estatisticamente comprovada em 23 das 28 funções, perdendo em apenas uma e empatando em 4. Vale ressaltar que além de obter melhores resultados em média, o L-CMA-DE ainda possui mais robustez nos resultados apresentando menores valores de desvios.

A Tabela 20 apresenta os resultados dos testes realizados entre o algoritmo proposto e o DE. Observa-se novamente a superioridade nos resultados obtidos com o L-CMA-DE frente ao *DE/rand/1/bin*. Os resultados comprovam estatisticamente o melhor desempenho do algoritmo memético proposto em 23 das 28 funções, perdendo em 2 e empatando em 3.

A Tabela 21 contém os resultados dos testes realizados entre o L-CMA-DE e o CMA-ES-RIS, apresentando um certo equilíbrio de resultados. Entretanto, o algoritmo proposto ganha em 12, perde em 10 e empada em 6 funções. Vale ressaltar que o CMA-ES-RIS apresenta 3 melhores resultados contra 2 do algoritmo proposto, quando se trata da classe de Funções Compostas. Isto sugere que criar algoritmos meméticos com balanço entre características explorativas e explotativas, é uma boa escolha para obter-se bons resultados.

A Tabela 22 apresenta os resultados dos testes realizados entre o algoritmo proposto e o IPOP-CMA-ES. Observa-se que o IPOP-CMA-ES apresenta melhores resultados nas funções menos complexas, ao passo que o algoritmo proposto apresenta superioridade em 6 dentre as 8 Funções Compostas existentes.

A Tabela 23 contém os resultados dos testes realizados entre o L-CMA-DE e o NBIPOP-aCMA-ES, que foi o algoritmo campeão da CEC-2013 e é reconhecido pelo seu bom desempenho em ambiente multimodais. Observa-se que o NBIPOP-aCMA-ES apresenta resultados superiores ao algoritmo proposto em 18 funções de teste, perdendo em 4 e empatando em 6 delas. Contudo, vale ressaltar que dentre as 4 funções que o L-CMA-DE apresentou melhores resultados, 3 delas são da classe de Funções Compostas e uma da classe de Funções Multimodais Básicas e possui diversos mínimos locais, o que evidencia o bom desempenho do algoritmo proposto nesta classe de funções.

Procurando evidenciar que o L-CMA-DE apresenta bons resultados quando aplicado às Funções Compostas, as Tabelas 24 a 28 contêm um recorte dos testes realizados com funções desta classe, apresentados ao longo desta seção.

4.8.2 Testes com as funções da CEC-2017

Para este conjunto de problemas utilizou-se 29 das 30 funções de teste³. Na Tabela 29 estão apresentados os testes realizados entre o algoritmo proposto e o CMA-ES. Os resultados ratificam a superioridade da nova proposta frente ao seu antecessor, obtendo os melhores médias em todas as funções testadas, com empate apenas nas funções 2 e 3.

A Tabela 30 apresenta os resultados dos testes realizados entre o algoritmo proposto e o DE, onde a nova proposta apresenta melhores resultados em 19 funções, perde em 6 e empata em 5 funções. Também é possível perceber que ao observar os melhores indivíduos encontrados, o algoritmo proposto também apresenta superioridade.

Na Tabela 31 pode ser visto os resultados dos testes realizados entre o L-CMA-DE e o RB-IPOP-CMA-ES, onde mais uma vez é perceptível a superioridade nos resultados alcançados pelo algoritmo proposto nas Funções Compostas, obtendo melhores médias em 7 das 8 funções. No que tange as funções unimodais, apenas houve diferença estatística na Função 1. Havendo queda de desempenho comparativo da proposta para o conjunto de funções multimodais que são híbridas.

Na Tabela 32 pode ser visto os resultados dos testes realizados entre o algoritmo proposto e o LSHADE-SPACMA, onde observa-se que a nova proposta continua obtendo melhores resultados no nicho das Funções Compostas (ganhou em 6 e perdeu em 4). Observando-se os melhores indivíduos encontrados, o algoritmo proposto obtém o ótimo global em 5 das 10 Funções Compostas, ao passo que o LSHADE-SPACMA não consegue encontrá-los nos testes executados.

Procurando mais uma vez evidenciar a boa qualidade dos resultados alcançados pelo L-CMA-DE em Funções Compostas, as Tabelas 33 a 35 contêm um recorte dos testes realizados estas funções, apresentados ao longo desta seção.

Podemos também observar o desempenho do L-CMA-DE no contexto de cada classe de função das competições, analisando a quantidade de vezes em que ele foi estatisticamente melhor pior ou igual, quando comparado aos algoritmos utilizados nos testes. Estas análises podem ser observadas nas Tabelas 36-44, onde os valores em negrito representam a maior quantidade de funções em que o algoritmo proposto foi estatisticamente melhor ou pior, nas comparações pareadas. Com esta análise, mais uma vez fica indicada a relevância dos resultados obtidos pelo L-CMA-DE frente outras propostas, no que diz respeito às funções mais difíceis.

A partir dos resultados obtidos com os testes executados e das análises realizadas nesta Seção, observa-se que o L-CMA-DE apresenta desempenho relevante na classe de Funções Complexas, mostrando-se como uma alternativa para solucionar problemas de otimização de alta multimodalidade, devido sua forma de agregar exploração e exploração durante do processo evolucionário.

³ A Função 2 (*Shifted and Rotated Sum of Different Power Function*) não foi avaliada devido à instabilidade observada e relatada em Awad et al. (2016). Foram inseridos valores nulos em suas métricas.

Tabela 19 – Testes com o algoritmo proposto e o CMA-ES (CEC-2013 / 10-D)

Função	L-CMA-DE				CMA-ES				Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN
2	586.1110000	770.1585000	618.5772000	2.0400000	0.0000000	1072783.0000000	5874535.0000000	0.0000000	0.0000000
3	0.0600000	0.5631373	1.5722170	0.0000000	0.0000000	7322829538.0000000	10760580943.0000000	0.0000000	0.4546442
4	761.8150000	881.4805000	523.7928000	63.0700000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
5	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	52.8543100	377.4553000	0.0000000	0.3268926
6	0.0000000	0.0000196	0.0001400	0.0000000	49.2380000	69.2573900	71.4098300	0.0000000	0.0000000
7	0.5900000	2.3920590	3.7784840	0.0040000	100.4990000	163.6222000	264.6503000	0.3010000	0.0000000
8	20.3470000	20.3383100	0.0844428	20.0000000	20.3670000	20.3465300	0.1071943	20.0000000	0.3504553
9	4.2570000	4.1175880	1.2765470	1.7190000	10.6390000	11.4399000	3.8068180	4.0810000	0.0000000
10	0.0000000	0.0000392	0.0002801	0.0000000	0.0120000	0.0146471	0.0126029	0.0000000	0.0000000
11	6.9650000	7.5111570	2.8857120	1.9900000	22.8840000	194.3707000	297.0651000	2.9850000	0.0000000
12	7.9600000	8.3695100	2.9858710	3.9800000	17.9090000	195.8902000	278.4561000	3.9800000	0.0000000
13	17.4410000	19.1185300	10.2137000	3.9730000	29.7140000	33.5192500	20.3521400	6.3000000	0.0000593
14	809.9240000	833.4357000	233.3787000	314.3570000	1828.8500000	1805.1650000	393.2520000	1043.4550000	0.0000000
15	758.7890000	750.0151000	196.8635000	236.4120000	1727.5700000	1719.7430000	527.4069000	373.5430000	0.0000000
16	0.0250000	0.0257451	0.0112745	0.0010000	0.1680000	0.1879412	0.1162534	0.0000000	0.0000000
17	17.5370000	19.3919600	6.5437260	9.8550000	645.9170000	590.3176000	482.3128000	11.9330000	0.0000000
18	17.8850000	18.2929800	3.9305640	11.3550000	550.0980000	594.5602000	506.4540000	9.8010000	0.0000000
19	0.4990000	0.5203922	0.1249522	0.2400000	1.0140000	4.2810000	22.8118600	0.4870000	0.0000000
20	3.2980000	3.2279410	0.3160663	2.0630000	4.9960000	4.8136670	0.3842934	2.9360000	0.0000000
21	200.0000000	219.0286000	125.7581000	100.0000000	400.1900000	351.1294000	98.8125900	100.0000000	0.0000002
22	810.0400000	807.6710000	189.2588000	370.8100000	2317.3500000	2300.5280000	427.7953000	1373.0200000	0.0000000
23	798.0900000	820.8102000	261.7469000	345.2300000	2388.6400000	2304.5650000	470.8843000	1146.7300000	0.0000000
24	131.0700000	137.8965000	26.6147300	103.8400000	347.2900000	369.9061000	155.2312000	105.9300000	0.0000000
25	211.0000000	203.4327000	22.1714800	115.4500000	232.3400000	255.8457000	47.3708700	202.1600000	0.0000000
26	107.9600000	108.7547000	8.3994840	66.1200000	211.1500000	255.6776000	132.1962000	103.9800000	0.0000000
27	300.3100000	315.1177800	38.3133000	234.4200000	649.4800000	565.6847000	171.7630000	300.0300000	0.0000000
28	100.0000000	110.2365000	40.9329900	100.0000000	565.7800000	1203.9580000	1220.7310000	100.0000000	0.0000000

Fonte: Próprio Autor

Tabela 20 – Testes com o algoritmo proposto e o DE (CEC-2013 / 10-D)

Função	L-CMA-DE					DE/rand/1/bin					Valor P		
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média		Desvio Padrão	Melhor Aptidão
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN
2	586.1110000	770.1585000	618.572000	2.040000	2.990000	2.9529410	1.2892980	0.560000	135580945.0000000	2.9529410	1.2892980	0.560000	0.000000
3	0.060000	0.5631373	1.5722170	0.000000	129363200.0000000	135580945.0000000	46983290.0000000	63477300.0000000	0.0100000	0.0056569	0.0000000	0.0000000	0.000000
4	761.8150000	881.4805000	523.7928000	63.070000	0.0100000	0.0100000	0.0056569	0.000000	0.0040000	0.0037255	0.0020000	0.0000000	0.000000
5	0.000000	0.0000000	0.000000	0.000000	0.0040000	0.0037255	0.0009182	0.000000	0.3610000	0.4139804	0.1200000	0.0000000	0.000000
6	0.000000	0.0000196	0.0001400	0.000000	0.3610000	0.4139804	0.2016768	0.000000	32.0960600	6.0760070	0.1200000	0.0000000	0.000000
7	0.590000	2.3920590	3.7784840	0.004000	32.0960600	6.0760070	6.0760070	17.4190000	20.3443100	0.0626640	20.1550000	0.8330182	0.000000
8	20.3470000	20.3383100	0.0844428	20.000000	20.3460000	20.3443100	0.0626640	20.1550000	8.3933530	0.7088265	5.9130000	0.0000000	0.000000
9	4.2570000	4.1175880	1.2765470	1.719000	8.428000	8.3933530	0.7088265	5.9130000	0.6173529	0.0882142	0.3440000	0.0000000	0.000000
10	0.000000	0.0000392	0.0002801	0.000000	0.6350000	0.6173529	0.0882142	0.3440000	42.4906500	5.8196700	27.2750000	0.0000000	0.000000
11	6.9650000	7.5111570	2.8857120	1.990000	43.510000	42.4906500	5.8196700	27.2750000	42.6451800	5.4133280	27.4570000	0.0000000	0.000000
12	7.960000	8.3695100	2.9858710	3.980000	43.019000	42.6451800	5.4133280	27.4570000	41.5818200	4.8587470	29.1520000	0.0000000	0.000000
13	17.4410000	19.1185300	10.2137000	3.973000	41.650000	41.5818200	4.8587470	29.1520000	1669.0060000	128.2316000	1269.7600000	0.0000000	0.000000
14	809.9240000	833.4357000	233.3787000	314.357000	1687.260000	1669.0060000	128.2316000	1269.7600000	1515.0620000	161.8740000	1010.1400000	0.0000000	0.000000
15	758.7890000	750.0151000	196.8635000	236.412000	1547.200000	1515.0620000	161.8740000	1010.1400000	1.1877250	0.1867412	0.6910000	0.0000000	0.000000
16	0.0250000	0.0257451	0.0112745	0.0010000	1.1930000	1.1877250	0.1867412	0.6910000	55.3864300	7.5708670	35.6380000	0.0000000	0.000000
17	17.5370000	19.3919600	6.5437260	9.855000	56.005000	55.3864300	7.5708670	35.6380000	58.3023900	5.6840390	44.6320000	0.0000000	0.000000
18	17.8850000	18.2929800	3.9305640	11.355000	59.086000	58.3023900	5.6840390	44.6320000	3.4068820	0.4567078	2.5300000	0.0000000	0.000000
19	0.4990000	0.5203922	0.1249522	0.240000	3.420000	3.4068820	0.4567078	2.5300000	3.5139020	0.1367413	3.1670000	0.0000000	0.000000
20	3.2980000	3.2279410	0.3160663	2.063000	3.548000	3.5139020	0.1367413	3.1670000	106.8365000	19.2639100	100.9770000	0.1755278	0.000000
21	200.0000000	219.0286000	125.7581000	100.000000	102.8870000	106.8365000	19.2639100	100.9770000	1477.8060000	154.3372000	945.1400000	0.0000000	0.000000
22	810.0400000	807.6710000	189.2588000	370.810000	1453.580000	1477.8060000	154.3372000	945.1400000	1592.7190000	192.9834000	897.0200000	0.0000000	0.000000
23	798.0900000	820.8102000	261.7469000	345.230000	1616.420000	1592.7190000	192.9834000	897.0200000	189.4690000	14.7107500	162.5800000	0.0000000	0.000000
24	131.0700000	137.8965000	26.6147300	103.840000	189.110000	189.4690000	14.7107500	162.5800000	218.5557000	9.0921580	175.4800000	0.0000000	0.000000
25	211.0000000	203.4327000	22.1714800	115.450000	221.270000	218.5557000	9.0921580	175.4800000	147.2375000	8.9441250	122.8300000	0.0000000	0.000000
26	107.9600000	108.7547000	8.3994840	66.120000	147.360000	147.2375000	8.9441250	122.8300000	549.1004000	27.0739100	471.8500000	0.0000000	0.000000
27	300.3100000	315.1778000	38.3133000	234.420000	555.960000	549.1004000	27.0739100	471.8500000	118.4600000	79.5635500	35.5300000	0.0000000	0.000000
28	100.0000000	110.2365000	40.9329900	100.000000	118.460000	118.4600000	79.5635500	35.5300000					

Fonte: Próprio Autor

Tabela 21 – Testes com o algoritmo proposto e o CMA-ES-RIS (CEC-2013 / 10-D)

Função	L-CMA-DE					CMA-ES-RIS					Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão			
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN		
2	586.111000	770.1585000	618.5772000	2.040000	0.000000	0.000000	0.000000	0.000000	0.000000		
3	0.060000	0.5631373	1.5722170	0.000000	0.000000	0.7042525	4.6116240	0.000000	0.000000		
4	761.8150000	881.4805000	523.7928000	63.070000	0.000000	0.000000	0.000000	0.000000	0.000000		
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN		
6	0.000000	0.000196	0.0001400	0.000000	0.0024222	1.0978330	2.8818090	0.000000	0.000000		
7	0.590000	2.3920590	3.7784840	0.004000	40.0841900	53.3219700	46.7502300	1.9906670	0.000000		
8	20.3470000	20.3383100	0.0844428	20.000000	20.3388100	20.3051600	0.1369676	20.000000	0.3202767		
9	4.2570000	4.1175880	1.2765470	1.719000	3.6335240	3.5880720	1.0439440	1.0998970	0.0338723		
10	0.000000	0.0000392	0.0002801	0.000000	0.0098573	0.0123599	0.0134592	0.000000	0.000000		
11	6.9650000	7.5111570	2.8857120	1.990000	2.9848770	3.5701880	1.4780830	0.000000	0.000000		
12	7.9600000	8.3695100	2.9858710	3.980000	12.9344600	12.8564200	5.4162340	1.9899180	0.000197		
13	17.4410000	19.1185300	10.2137000	3.973000	26.3701600	25.5660400	10.8190000	4.4814380	0.0027748		
14	809.9240000	833.4357000	233.3787000	314.3570000	113.2689000	101.9323000	73.8826500	3.6023240	0.000000		
15	758.7890000	750.0151000	196.8635000	236.4120000	633.0726000	617.4359000	173.9748000	142.0349000	0.0005010		
16	0.0250000	0.0257451	0.0112745	0.001000	0.1758882	0.1642744	0.0756188	0.000000	0.000000		
17	17.5370000	19.3919600	6.5437260	9.855000	11.8298900	10.3770200	3.7341610	1.6856280	0.000000		
18	17.8850000	18.2929800	3.9305640	11.3550000	29.4695800	29.7762600	6.1606130	19.4808700	0.000000		
19	0.4990000	0.5203922	0.1249522	0.240000	0.7607153	0.8143309	0.2740658	0.3944331	0.000000		
20	3.2980000	3.2279410	0.3160663	2.063000	4.2534510	4.1599670	0.3985509	3.3467990	0.000000		
21	200.0000000	219.0286000	125.7581000	100.000000	200.0000000	160.7881000	60.2762700	100.000000	0.1063125		
22	810.0400000	807.6710000	189.2588000	370.8100000	268.4397000	243.8627000	108.5685000	50.4929000	0.000000		
23	798.0900000	820.8102000	261.7469000	345.2300000	833.7798000	834.8106000	190.4691000	339.5744000	0.4535058		
24	131.0700000	137.8965000	26.6147300	103.8400000	118.1697000	118.5079000	5.6885290	106.5825000	0.0000279		
25	211.0000000	203.4327000	22.1714800	115.4500000	207.3190000	192.6559000	34.2264300	113.8599000	0.1082180		
26	107.9600000	108.7547000	8.3994840	66.1200000	200.0174000	161.0954000	40.5879700	103.9798000	0.000000		
27	300.3100000	315.1778000	38.3133000	234.4200000	307.7429000	312.8075000	22.9558700	225.1812000	0.0000131		
28	100.0000000	110.2365000	40.9329900	100.0000000	300.0000000	205.8824000	106.6054000	0.0000017	0.0000244		

Fonte: Próprio Autor

Tabela 22 – Testes com o algoritmo proposto e o IPOP-CMA-ES (CEC-2013 / 10-D)

Função	L-CMA-DE				IPOP-CMA-ES				Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN
2	586.1110000	770.1585000	618.5772000	2.040000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.060000	0.5631373	1.5722170	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	761.8150000	881.4805000	523.7928000	63.070000	0.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN
6	0.000000	0.000196	0.000140	0.000000	0.000000	0.000000	0.000000	0.000000	0.3268926
7	0.590000	2.3920590	3.7784840	0.004000	0.000000	0.0000005	0.000014	0.000000	0.000000
8	20.3470000	20.3383100	0.0844428	20.000000	20.3520000	20.3417800	0.0697129	20.1870000	0.9439719
9	4.2570000	4.1175880	1.2765470	1.7190000	0.1990700	0.5822977	0.7212153	0.0000000	0.0000000
10	0.000000	0.0000392	0.0002801	0.000000	0.000000	0.000000	0.000000	0.000000	0.3268926
11	6.9650000	7.5111570	2.8857120	1.9900000	0.000000	0.3316525	0.5509815	0.0000000	0.0000000
12	7.9600000	8.3695100	2.9858710	3.9800000	0.000000	0.0975451	0.2988130	0.0000000	0.0000000
13	17.4410000	19.1185300	10.2137000	3.9730000	0.000000	0.3127619	0.5078043	0.0000000	0.0000000
14	809.9240000	833.4357000	233.3787000	314.3570000	18.5350000	26.8787500	26.7365000	3.6023000	0.0000000
15	758.7890000	750.0151000	196.8635000	236.4120000	18.5350000	21.9549400	15.2425700	0.3122700	0.0000000
16	0.0250000	0.0257451	0.0112745	0.0010000	1.1241000	1.1518950	0.1364718	0.9052100	0.0000000
17	17.5370000	19.3919600	6.5437260	9.8550000	10.9840000	11.0684700	0.4290930	10.3820000	0.0000000
18	17.8850000	18.2929800	3.9305640	11.3550000	10.9510000	10.9770600	0.4115409	10.2580000	0.0000000
19	0.4990000	0.5203922	0.1249522	0.2400000	0.6459300	0.6459273	0.1153117	0.4400500	0.0000043
20	3.2980000	3.2279410	0.3160663	2.0630000	3.0192000	2.7635690	0.5915633	1.5468000	0.0000026
21	200.0000000	219.0286000	125.7581000	100.0000000	400.1900000	374.6775000	71.7354600	100.0000000	0.0000000
22	810.0400000	807.6710000	189.2588000	370.8100000	59.5470000	73.3805700	49.4390700	9.9023000	0.0000000
23	798.0900000	820.8102000	261.7469000	345.2300000	59.2120000	86.1636500	66.0621900	14.2240000	0.0000000
24	131.0700000	137.8965000	26.6147300	103.8400000	208.4500000	209.4645000	7.0220600	200.0000000	0.0000000
25	211.0000000	203.4327000	22.1714800	115.4500000	203.6100000	205.5169000	6.7086260	200.0000000	0.0020094
26	107.9600000	108.7547000	8.3994840	66.1200000	205.8100000	204.2010000	15.0939200	106.9600000	0.0000000
27	300.3100000	315.1778000	38.3133000	234.4200000	446.6300000	454.5620000	79.2343300	319.7000000	0.0000000
28	100.0000000	110.2365000	40.9329900	100.0000000	300.0000000	300.0000000	0.0000000	300.0000000	0.0000000

Fonte: Próprio Autor

Tabela 23 – Testes com o algoritmo proposto e o NBIPOP-aCMA-ES (CEC-2013 / 10-D)

Função	L-CMA-DE					NBIPOP-aCMA-ES					Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	Valor P		
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN		
2	586.111000	770.1585000	618.5772000	2.040000	0.000000	0.000000	0.000000	0.000000	0.000000		
3	0.060000	0.5631373	1.5722170	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000		
4	761.8150000	881.4805000	523.7928000	63.070000	0.000000	0.000000	0.000000	0.000000	0.000000		
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN		
6	0.000000	0.0000196	0.0001400	0.000000	0.000000	0.000000	0.000000	0.000000	0.3268926		
7	0.5900000	2.3920590	3.7784840	0.004000	0.000000	0.0000026	0.0000053	0.000000	0.000000		
8	20.3470000	20.3383100	0.0844428	20.000000	20.3530000	20.3385300	0.0897736	20.000000	0.9973298		
9	4.2570000	4.1175880	1.2765470	1.719000	0.000000	0.2317159	0.4395774	0.000000	0.000000		
10	0.000000	0.0000392	0.0002801	0.000000	0.000000	0.000000	0.000000	0.000000	0.3268926		
11	6.9650000	7.5111570	2.8857120	1.990000	0.0002850	0.3686430	0.5091779	0.000000	0.000000		
12	7.9600000	8.3695100	2.9858710	3.980000	0.000000	0.2379207	0.5422559	0.000000	0.000000		
13	17.4410000	19.1185300	10.2137000	3.973000	0.0012640	0.4840336	0.6758130	0.000000	0.000000		
14	809.9240000	833.4357000	233.3787000	314.3570000	76.8160000	113.9740000	91.4690200	6.8924000	0.000000		
15	758.7890000	750.0151000	196.8635000	236.4120000	151.3100000	158.1614000	117.3167000	18.5970000	0.000000		
16	0.0250000	0.0257451	0.0112745	0.001000	0.0536430	0.1196388	0.2633395	0.0105870	0.0000158		
17	17.5370000	19.3919600	6.5437260	9.855000	11.3690000	11.3398200	0.5563440	10.3330000	0.000000		
18	17.8850000	18.2929800	3.9305640	11.355000	11.0710000	11.2885100	1.2760820	7.9559000	0.000000		
19	0.4990000	0.5203922	0.1249522	0.240000	0.5181800	0.5251880	0.1388111	0.0098770	0.7480196		
20	3.2980000	3.2279410	0.3160663	2.063000	2.7612000	2.7259610	0.6495309	1.1977000	0.0000544		
21	200.0000000	219.0286000	125.7581000	100.000000	200.0000000	152.9412000	50.4100800	100.000000	0.0336821		
22	810.0400000	807.6710000	189.2588000	370.8100000	141.8200000	175.1307000	114.6550000	36.3550000	0.000000		
23	798.0900000	820.8102000	261.7469000	345.2300000	129.1800000	174.2297000	122.8315000	24.4530000	0.000000		
24	131.0700000	137.8965000	26.6147300	103.8400000	107.8700000	119.8851000	32.2201500	100.000000	0.0000001		
25	211.0000000	203.4327000	22.1714800	115.4500000	200.0600000	176.9724000	39.9180600	100.000000	0.000000		
26	107.9600000	108.7547000	8.3994840	66.1200000	105.9700000	111.0351000	24.9864000	100.000000	0.0036320		
27	300.3100000	315.1778000	38.3133000	234.4200000	311.6200000	316.6843000	29.5556200	172.9800000	0.0107467		
28	100.0000000	110.2365000	40.9329900	100.0000000	300.0000000	249.0196000	88.0285200	100.0000000	0.0000000		

Fonte: Próprio Autor

Tabela 24 – Testes com o algoritmo proposto e o CMA-ES (CEC-2013 / Funções Compostas / 10-D)

Função	L-CMA-DE					CMA-ES					Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Melhor Aptidão	
21	200.0000000	219.0286000	125.7581000	100.0000000	400.1900000	351.1294000	98.8125900	100.0000000	100.0000000	100.0000000	0.0000002
22	810.0400000	807.6710000	189.2588000	370.8100000	2317.3500000	2300.5280000	427.7953000	1373.0200000	1373.0200000	1373.0200000	0.0000000
23	798.0900000	820.8102000	261.7469000	345.2300000	2388.6400000	2304.5650000	470.8843000	1146.7300000	1146.7300000	1146.7300000	0.0000000
24	131.0700000	137.8965000	26.6147300	103.8400000	347.2900000	369.9061000	155.2312000	105.9300000	105.9300000	105.9300000	0.0000000
25	211.0000000	203.4327000	22.1714800	115.4500000	232.3400000	255.8457000	47.3708700	202.1600000	202.1600000	202.1600000	0.0000000
26	107.9600000	108.7547000	8.3994840	66.1200000	211.1500000	255.6776000	132.1962000	103.9800000	103.9800000	103.9800000	0.0000000
27	300.3100000	315.1778000	38.3133000	234.4200000	649.4800000	565.6847000	171.7630000	300.0300000	300.0300000	300.0300000	0.0000000
28	100.0000000	110.2365000	40.9329900	100.0000000	565.7800000	1203.9580000	1220.7310000	100.0000000	100.0000000	100.0000000	0.0000000

Fonte: Próprio Autor

Tabela 25 – Testes com o algoritmo proposto e o DE (CEC-2013 / Funções Compostas / 10-D)

Função	L-CMA-DE					DE/rand/1/bin					Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Melhor Aptidão	
21	200.0000000	219.0286000	125.7581000	100.0000000	102.8870000	106.8365000	19.2639100	100.9770000	100.9770000	0.1755278	
22	810.0400000	807.6710000	189.2588000	370.8100000	1453.5800000	1477.8060000	154.3372000	945.1400000	945.1400000	0.0000000	
23	798.0900000	820.8102000	261.7469000	345.2300000	1616.4200000	1592.7190000	192.9834000	897.0200000	897.0200000	0.0000000	
24	131.0700000	137.8965000	26.6147300	103.8400000	189.1100000	189.4690000	14.7107500	162.5800000	162.5800000	0.0000000	
25	211.0000000	203.4327000	22.1714800	115.4500000	221.2700000	218.5557000	9.0921580	175.4800000	175.4800000	0.0000000	
26	107.9600000	108.7547000	8.3994840	66.1200000	147.3600000	147.2375000	8.9441250	122.8300000	122.8300000	0.0000000	
27	300.3100000	315.1778000	38.3133000	234.4200000	555.9600000	549.1004000	27.0739100	471.8500000	471.8500000	0.0000000	
28	100.0000000	110.2365000	40.9329900	100.0000000	118.4600000	155.3508000	79.5635500	35.5300000	35.5300000	0.0000000	

Fonte: Próprio Autor

Tabela 26 – Testes com o algoritmo proposto e o CMA-ES-RIS (CEC-2013 / Funções Compostas / 10-D)

Função	L-CMA-DE				CMA-ES-RIS				Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
21	200.000000	219.0286000	125.7581000	100.0000000	200.0000000	160.7881000	60.2762700	100.0000000	0.1063125
22	810.040000	807.6710000	189.2588000	370.8100000	268.4397000	243.8627000	108.5685000	50.4929000	0.0000000
23	798.090000	820.8102000	261.7469000	345.2300000	833.7798000	834.8106000	190.4691000	339.5744000	0.4535058
24	131.070000	137.8965000	26.6147300	103.8400000	118.1697000	118.5079000	5.6885290	106.5825000	0.0000279
25	211.000000	203.4327000	22.1714800	115.4500000	207.3190000	192.6559000	34.2264300	113.8599000	0.1082180
26	107.960000	108.7547000	8.3994840	66.1200000	200.0174000	161.0954000	40.5879700	103.9798000	0.0000000
27	300.310000	315.1778000	38.3133000	234.4200000	307.7429000	312.8075000	22.9558700	225.1812000	0.0000131
28	100.000000	110.2365000	40.9329900	100.0000000	300.0000000	205.8824000	106.6054000	0.0000017	0.0000244

Fonte: Próprio Autor

Tabela 27 – Testes com o algoritmo proposto e o IPOP-CMA-ES (CEC-2013 / Funções Compostas / 10-D)

Função	L-CMA-DE				IPOP-CMA-ES				Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
21	200.0000000	219.0286000	125.7581000	100.0000000	400.1900000	374.6775000	71.7354600	100.0000000	0.0000000
22	810.0400000	807.6710000	189.2588000	370.8100000	59.5470000	73.3805700	49.4390700	9.9023000	0.0000000
23	798.0900000	820.8102000	261.7469000	345.2300000	59.2120000	86.1636500	66.0621900	14.2240000	0.0000000
24	131.0700000	137.8965000	26.6147300	103.8400000	208.4500000	209.4645000	7.0220600	200.0000000	0.0000000
25	211.0000000	203.4327000	22.1714800	115.4500000	203.6100000	205.5169000	6.7086260	200.0000000	0.0020094
26	107.9600000	108.7547000	8.3994840	66.1200000	205.8100000	204.2010000	15.0939200	106.9600000	0.0000000
27	300.3100000	315.1778000	38.3133000	234.4200000	446.6300000	454.5620000	79.2343300	319.7000000	0.0000000
28	100.0000000	110.2365000	40.9329900	100.0000000	300.0000000	300.0000000	0.0000000	300.0000000	0.0000000

Fonte: Próprio Autor

Tabela 28 – Testes com o algoritmo proposto e o NBIPOP-aCMA-ES (CEC-2013 / Funções Compostas / 10-D)

Função	L-CMA-DE				NBIPOP-aCMA-ES				Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
21	200.0000000	219.0286000	125.7581000	100.0000000	200.0000000	152.9412000	50.4100800	100.0000000	0.0336821
22	810.0400000	807.6710000	189.2588000	370.8100000	141.8200000	175.1307000	114.6550000	36.3550000	0.0000000
23	798.0900000	820.8102000	261.7469000	345.2300000	129.1800000	174.2297000	122.8315000	24.4530000	0.0000000
24	131.0700000	137.8965000	26.6147300	103.8400000	107.8700000	119.8851000	32.2201500	100.0000000	0.0000001
25	211.0000000	203.4327000	22.1714800	115.4500000	200.0600000	176.9724000	39.9180600	100.0000000	0.0000000
26	107.9600000	108.7547000	8.3994840	66.1200000	105.9700000	111.0351000	24.9864000	100.0000000	0.0036320
27	300.3100000	315.1778000	38.3133000	234.4200000	311.6200000	316.6843000	29.5556200	172.9800000	0.0107467
28	100.0000000	110.2365000	40.9329900	100.0000000	300.0000000	249.0196000	88.0285200	100.0000000	0.0000000

Fonte: Próprio Autor

Tabela 29 – Testes com o algoritmo proposto e o CMA-ES (CEC-2017 / 10-D)

Função	L-CMA-DE				CMA-ES				Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
1	0,0060000	0,0370000	0,0961362	0,0000000	0,0000000	533910618,0000000	1608695595,0000000	0,0000000	0,0000000
2	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	NaN
3	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	NaN
4	0,0000000	0,0000000	0,0000000	0,0000000	0,0000000	44,6782900	127,6868000	0,0000000	0,0018383
5	10,9450000	11,6282200	4,0939110	2,9850000	215,9020000	222,3110000	97,3529800	5,9700000	0,0000000
6	10,2420000	10,1271400	5,5416170	0,0400000	88,6350000	89,2462400	17,4189200	59,5250000	0,0000000
7	23,9360000	24,1360200	7,4649990	9,0140000	854,7300000	838,1878000	417,9105000	15,6440000	0,0000000
8	13,9290000	14,8071800	5,2332270	4,9750000	150,2320000	155,2298000	70,8250400	38,8030000	0,0000000
9	173,2900000	188,2392000	86,3196100	0,0000000	3628,7300000	3941,5850000	1865,2970000	365,5200000	0,0000000
10	664,1500000	672,0198000	170,2967000	200,8500000	1707,1200000	1708,5580000	423,8994000	827,6200000	0,0000000
11	12,6600000	13,2086300	6,1960180	1,9900000	71,6400000	113,0418000	300,4616000	8,9500000	0,0000000
12	236,8800000	223,9622000	129,2685000	0,0000000	578,0700000	17050851,0000000	69462054,0000000	11,1700000	0,0000000
13	78,3000000	95,4219600	63,0885200	14,6800000	281,8200000	332,1067000	221,5550000	63,1000000	0,0000000
14	27,0300000	28,0360800	5,2350690	11,6400000	63,7800000	92,4456900	102,4451000	26,9600000	0,0000000
15	34,8400000	34,9627500	16,3579400	6,8300000	86,0400000	211,6998000	644,9876000	4,0600000	0,0000000
16	3,3500000	8,0335290	17,4053900	1,3100000	406,8900000	414,4282000	214,9505000	1,4300000	0,0000000
17	36,9500000	36,0056900	9,2830610	19,2700000	128,1800000	202,6527000	172,0620000	24,0100000	0,0000000
18	38,9100000	42,7582400	16,7266200	19,7500000	252,7200000	1958410,0000000	4036081,0000000	21,1600000	0,0000000
19	11,8000000	13,9415700	8,1328000	3,9300000	56,4500000	66128,5200000	245068,9000000	7,3500000	0,0000000
20	53,9000000	55,1007800	17,5505300	21,9900000	511,0600000	501,0186000	207,2167000	163,4000000	0,0000000
21	100,0000000	100,0012000	0,0084017	100,0000000	234,2800000	266,1663000	116,4010000	100,0000000	0,0000000
22	100,0000000	73,6227500	33,8581100	0,0000000	1906,4100000	1834,1970000	859,7388000	19,2200000	0,0000000
23	311,9100000	311,4549000	3,1097380	304,0700000	774,1000000	882,5884000	471,7632000	306,0700000	0,0000000
24	100,0000000	102,0606000	24,4169500	0,0000000	413,3100000	341,4249000	133,1909000	100,0000000	0,0000000
25	397,7400000	359,9831000	93,8470700	0,0000000	443,5000000	438,6733000	40,4419800	397,7500000	0,0000000
26	200,0000000	207,8431000	95,5889900	0,0000000	774,6500000	1092,1580000	881,4099000	200,0000000	0,0000000
27	389,0800000	389,2237000	0,5055688	387,3200000	430,8700000	552,1373000	534,1504000	448,6700000	0,0000000
28	300,0000000	290,1961000	50,0196000	0,0000000	627,6600000	643,2747000	186,5152000	300,0000000	0,0000000
29	246,4700000	248,4431000	11,8854200	204,9900000	379,3200000	441,0016000	181,2515000	257,2700000	0,0000000
30	504,8400000	540,0845000	119,2817000	400,1000000	882863,0000000	805035,4000000	1082058,0000000	882863,0000000	0,0000738

Fonte: Próprio Autor

Tabela 30 – Testes com o algoritmo proposto e o DE (CEC-2017 / 10-D)

Função	L-CMA-DE				DE/rand/1/bin				Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
1	0.0060000	0.0370000	0.0961362	0.0000000	106.3030000	120.4778000	55.7304100	37.3600000	0.0000000
2	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	NaN
3	0.0000000	0.0000000	0.0000000	0.0000000	0.0010000	0.0011176	0.0004311	0.0000000	0.0000000
4	0.0000000	0.0000000	0.0000000	0.0000000	0.6940000	0.7087255	0.2395382	0.2290000	0.0000000
5	10.9450000	11.6282200	4.0939110	2.9850000	38.4020000	38.2456900	5.3208460	26.1000000	0.0000000
6	10.2420000	10.1271400	5.5416170	0.0400000	10.1710000	10.2626900	1.3457310	7.7390000	0.9413128
7	23.9360000	24.1360200	7.4649990	9.0140000	57.4440000	57.3660000	5.1328880	44.9780000	0.0000000
8	13.9290000	14.8071800	5.2332270	4.9750000	44.1070000	43.6151200	4.8867150	31.4040000	0.0000000
9	173.2900000	188.2392000	86.3196100	0.0000000	86.8100000	90.2011000	26.5685800	36.8800000	0.0000000
10	664.1500000	672.0198000	170.2967000	200.8500000	1020.0700000	999.8184000	157.6632000	619.4200000	0.0000000
11	12.6600000	13.2086300	6.1960180	1.9900000	15.9700000	15.6211800	2.9545430	7.9300000	0.0019862
12	236.8800000	223.9622000	129.2685000	0.0000000	220.7800000	218.5761000	53.4947000	83.4700000	0.3522174
13	78.3000000	95.4219600	63.0885200	14.6800000	20.9200000	20.9443100	2.8433480	13.2900000	0.0000000
14	27.0300000	28.0360800	5.2350690	11.6400000	25.8700000	25.2886300	2.4978340	14.6300000	0.0020794
15	34.8400000	34.9627500	16.3579400	6.8300000	7.8000000	7.7617650	1.2819010	4.6800000	0.0000000
16	3.3500000	8.0335290	17.4053900	1.3100000	40.9700000	42.3080400	12.1267200	20.6300000	0.0000000
17	36.9500000	36.0056900	9.2830610	19.2700000	63.4300000	62.5819600	8.4354070	41.0000000	0.0000000
18	38.9100000	42.7582400	16.7266200	19.7500000	19.0500000	18.7566700	3.0758240	11.4000000	0.0000000
19	11.8000000	13.9415700	8.1328000	3.9300000	4.5800000	4.5649020	0.4792468	3.3600000	0.0000000
20	53.9000000	55.1007800	17.5505300	21.9900000	73.5300000	73.0452900	7.7008130	55.1600000	0.0000000
21	100.0000000	100.0012000	0.0084017	100.0000000	101.7700000	102.4216000	2.1212450	100.7800000	0.0000000
22	100.0000000	73.6227500	33.8581100	0.0000000	41.8400000	60.2021600	38.3683000	3.9900000	0.7917298
23	311.9100000	311.4549000	3.1097380	304.0700000	335.4400000	335.6620000	4.7210100	326.2200000	0.0000000
24	100.0000000	102.0606000	24.4169500	0.0000000	198.6300000	222.5284000	61.7210000	157.2400000	0.0000000
25	397.7400000	359.9831000	93.8470700	0.0000000	398.1500000	332.9043000	89.0647400	157.2600000	0.1234365
26	200.0000000	207.8431000	95.5889900	0.0000000	300.2800000	240.0784000	99.1822300	2.7100000	0.0000001
27	389.0800000	389.2237000	0.5055688	387.3200000	391.3000000	391.2216000	0.5750700	390.0600000	0.0000000
28	300.0000000	290.1961000	50.0196000	0.0000000	308.5500000	310.5912000	9.0610820	269.4000000	0.0000000
29	246.4700000	248.4431000	11.8854200	204.9900000	280.9700000	279.9565000	11.2734200	252.8100000	0.0000000
30	504.8400000	540.0845000	119.2817000	400.1000000	1099.1800000	1301.2760000	636.7207000	517.3400000	0.0000000

Fonte: Próprio Autor

Tabela 31 – Testes com o algoritmo proposto e o RB-IPOP-CMA-ES (CEC-2017 / 10-D)

Função	L-CMA-DE				RB-IPOP-CMA-ES				Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
1	0.0060000	0.0370000	0.0961362	0.0000000	0.0000000	0.0000000	0.0961362	0.0000000	0.0000000
2	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	NaN
3	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	NaN
4	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	NaN
5	10.9450000	11.6282200	4.0939110	2.9850000	0.9949591	1.5802290	4.0939110	0.0000000	0.0000000
6	10.2420000	10.1271400	5.5416170	0.0400000	0.0000001	0.0000002	5.5416170	0.0000000	0.0000000
7	23.9360000	24.1360200	7.4649990	9.0140000	10.8154200	10.1103600	7.4649990	0.0000006	0.0000000
8	13.9290000	14.8071800	5.2332270	4.9750000	0.9949591	1.9704080	5.2332270	0.0000000	0.0000000
9	173.2900000	188.2392000	86.3196100	0.0000000	0.0000000	0.0000000	86.3196100	0.0000000	0.0000000
10	664.1500000	672.0198000	170.2967000	200.8500000	407.2566000	435.4251000	170.2967000	0.3747266	0.0000000
11	12.6600000	13.2086300	6.1960180	1.9900000	0.0000000	0.1711793	6.1960180	0.0000000	0.0000000
12	236.8800000	223.9622000	129.2685000	0.0000000	119.9460000	110.4614000	129.2685000	0.2081430	0.0000059
13	78.3000000	95.4219600	63.0885200	14.6800000	4.8371370	4.1702330	63.0885200	0.0000000	0.0000000
14	27.0300000	28.0360800	5.2350690	11.6400000	20.0077800	15.9325000	5.2350690	0.0000001	0.0000000
15	34.8400000	34.9627500	16.3579400	6.8300000	0.4999946	0.4909027	16.3579400	0.0006506	0.0000000
16	3.3500000	8.0335290	17.4053900	1.3100000	119.1395000	97.0977400	17.4053900	0.6550901	0.0842156
17	36.9500000	36.0056900	9.2830610	19.2700000	41.7517400	52.4610200	9.2830610	6.6927427	0.1409141
18	38.9100000	42.7582400	16.7266200	19.7500000	20.5111100	19.7224800	16.7266200	0.0264224	0.0000000
19	11.8000000	13.9415700	8.1328000	3.9300000	1.2091260	1.8157840	8.1328000	0.0194319	0.0000000
20	53.9000000	55.1007800	17.5505300	21.9900000	138.4709000	106.0455000	17.5505300	1.6193254	0.0035990
21	100.0000000	100.0012000	0.0084017	100.0000000	100.0000000	137.3509000	0.0084017	100.0000000	0.0000000
22	100.0000000	73.6227500	33.8581100	0.0000000	100.0000000	99.2594800	33.8581100	60.2735149	0.0000000
23	311.9100000	311.4549000	3.1097380	304.0700000	302.8657000	275.0245000	3.1097380	100.0000004	0.0000000
24	100.0000000	102.0606000	24.4169500	0.0000000	200.0000000	197.5780000	24.4169500	0.0000016	0.0000000
25	397.7400000	359.9831000	93.8470700	0.0000000	398.0358000	402.2698000	93.8470700	100.0037924	0.0000058
26	200.0000000	207.8431000	95.5889900	0.0000000	300.0000000	272.6912000	95.5889900	0.0000011	0.0000000
27	389.0800000	389.2237000	0.5055688	387.3200000	395.0110000	394.8626000	0.5055688	393.8187647	0.0000000
28	300.0000000	290.1961000	50.0196000	0.0000000	300.0000000	402.2597000	50.0196000	0.0000003	0.0000000
29	246.4700000	248.4431000	11.8854200	204.9900000	254.4409000	265.8314000	11.8854200	226.2539544	0.3026923
30	504.8400000	540.0845000	119.2817000	400.1000000	442.6614000	2045.9850000	119.2817000	394.6131459	0.0023254

Fonte: Próprio Autor

Tabela 32 – Testes com o algoritmo proposto e o LSHADE-SPACMA (CEC-2017 / 10-D)

Função	L-CMA-DE				LSHADE-SPACMA				Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
1	0.0060000	0.0370000	0.0961362	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	NaN
3	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	NaN
4	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	NaN
5	10.9450000	11.6282200	4.0939110	2.9850000	0.9997125	1.3485700	0.7133901	0.0000026	0.0000000
6	10.2420000	10.1271400	5.5416170	0.0400000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
7	23.9360000	24.1360200	7.4649990	9.0140000	10.9250300	10.9541000	0.3509627	10.3673440	0.0000000
8	13.9290000	14.8071800	5.2332270	4.9750000	0.9953639	1.0358260	0.7438044	0.0000000	0.0000000
9	173.2900000	188.2392000	86.3196100	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
10	664.1500000	672.0198000	170.2967000	200.8500000	3.7487800	10.5115300	23.9905200	0.1954659	0.0000000
11	12.6600000	13.2086300	6.1960180	1.9900000	0.0000000	0.0195090	0.1393221	0.0000000	0.0000000
12	236.8800000	223.9622000	129.2685000	0.0000000	119.9460000	113.6777000	65.7041000	0.0000000	0.0000004
13	78.3000000	95.4219600	63.0885200	14.6800000	4.8371370	3.3696960	2.4171120	0.0000000	0.0000000
14	27.0300000	28.0360800	5.2350690	11.6400000	0.0000000	0.0531392	0.1952920	0.0000000	0.0000000
15	34.8400000	34.9627500	16.3579400	6.8300000	0.4845568	0.4112424	0.2659591	0.0031939	0.0000000
16	3.3500000	8.0335290	17.4053900	1.3100000	0.6987231	0.6419951	0.2295147	0.1959026	0.0000000
17	36.9500000	36.0056900	9.2830610	19.2700000	0.3323048	1.5429330	4.4391740	0.0197458	0.0000000
18	38.9100000	42.7582400	16.7266200	19.7500000	0.4683168	1.9742500	5.4430600	0.0115610	0.0000000
19	11.8000000	13.9415700	8.1328000	3.9300000	0.0199955	0.0812733	0.2077154	0.0000000	0.0000000
20	53.9000000	55.1007800	17.5505300	21.9900000	0.3121733	0.1836313	0.1672558	0.0000000	0.0000000
21	100.0000000	100.0012000	0.0084017	100.0000000	137.9377000	153.6940000	49.3101600	100.0000000	0.0000000
22	100.0000000	73.6227500	33.8581100	0.0000000	100.0000000	100.0191000	0.0781715	100.0000000	0.0000001
23	311.9100000	311.4549000	3.1097380	304.0700000	302.8729000	302.0450000	1.4971360	300.0011100	0.0000000
24	100.0000000	102.0606000	24.4169500	0.0000000	329.8444000	292.3530000	82.8574300	100.0000000	0.0000000
25	397.7400000	359.9831000	93.8470700	0.0000000	443.3300000	425.6128000	22.4131600	397.7428700	0.0000000
26	200.0000000	207.8431000	95.5889900	0.0000000	300.0000000	300.0000000	0.0000000	300.0000000	0.0000000
27	389.0800000	389.2237000	0.5055688	387.3200000	388.4065000	388.2803000	3.1282120	373.0700900	0.0000000
28	300.0000000	290.1961000	50.0196000	0.0000000	300.0000000	362.5444000	82.8609900	300.0000000	0.0000009
29	246.4700000	248.4431000	11.8854200	204.9900000	230.3615000	230.4884000	2.2398930	226.9131600	0.0000000
30	504.8400000	540.0845000	119.2817000	400.1000000	217.1836000	229.5842000	32.5351600	202.0146900	0.0000000

Fonte: Próprio Autor

Tabela 33 – Testes com o algoritmo proposto e o DE (CEC-2017 / Funções Compostas / 10-D)

Função	L-CMA-DE				DE/rand/1/bin				Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
21	100.0000000	100.0012000	0.0084017	100.0000000	101.7700000	102.4216000	2.1212450	100.7800000	0.0000000
22	100.0000000	73.6227500	33.8581100	0.0000000	41.8400000	60.2021600	38.3683000	3.9900000	0.7917298
23	311.9100000	311.4549000	3.1097380	304.0700000	335.4400000	335.6620000	4.7210100	326.2200000	0.0000000
24	100.0000000	102.0606000	24.4169500	0.0000000	198.6300000	222.5284000	61.7210000	157.2400000	0.0000000
25	397.7400000	359.9831000	93.8470700	0.0000000	398.1500000	332.9043000	89.0647400	157.2600000	0.1234365
26	200.0000000	207.8431000	95.5889900	0.0000000	300.2800000	240.0784000	99.1822300	2.7100000	0.0000001
27	389.0800000	389.2237000	0.5055688	387.3200000	391.3000000	391.2216000	0.5750700	390.0600000	0.0000000
28	300.0000000	290.1961000	50.0196000	0.0000000	308.5500000	310.5912000	9.0610820	269.4000000	0.0000000
29	246.4700000	248.4431000	11.8854200	204.9900000	280.9700000	279.9565000	11.2734200	252.8100000	0.0000000
30	504.8400000	540.0845000	119.2817000	400.1000000	1099.1800000	1301.2760000	636.7207000	517.3400000	0.0000000

Fonte: Próprio Autor

Tabela 34 – Testes com o algoritmo proposto e o RB-IPOP-CMA-ES (CEC-2017 / Funções Compostas / 10-D)

Função	L-CMA-DE				RB-IPOP-CMA-ES				Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	
21	100.000000	100.0012000	0.0084017	100.0000000	100.0000000	137.3509000	0.0084017	100.0000000	0.0000000
22	100.0000000	73.6227500	33.8581100	0.0000000	100.0000000	99.2594800	33.8581100	60.2735149	0.0000000
23	311.9100000	311.4549000	3.1097380	304.0700000	302.8657000	275.0245000	3.1097380	100.0000004	0.0000000
24	100.0000000	102.0606000	24.4169500	0.0000000	200.0000000	197.5780000	24.4169500	0.0000016	0.0000000
25	397.7400000	359.9831000	93.8470700	0.0000000	398.0358000	402.2698000	93.8470700	100.0037924	0.0000058
26	200.0000000	207.8431000	95.5889900	0.0000000	300.0000000	272.6912000	95.5889900	0.0000011	0.0000000
27	389.0800000	389.2237000	0.5055688	387.3200000	395.0110000	394.8626000	0.5055688	393.8187647	0.0000000
28	300.0000000	290.1961000	50.0196000	0.0000000	300.0000000	402.2597000	50.0196000	0.0000003	0.0000000
29	246.4700000	248.4431000	11.8854200	204.9900000	254.4409000	265.8314000	11.8854200	226.2539544	0.3026923
30	504.8400000	540.0845000	119.2817000	400.1000000	442.6614000	2045.9850000	119.2817000	394.6131459	0.0023254

Fonte: Próprio Autor

Tabela 35 – Testes com o algoritmo proposto e o LSHADE-SPACMA (CEC-2017 / Funções Compostas / 10-D)

Função	L-CMA-DE					LSHADE-SPACMA					Valor P
	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Média	Desvio Padrão	Melhor Aptidão	Mediana	Melhor Aptidão	
21	100.000000	100.0012000	0.0084017	100.0000000	137.9377000	153.6940000	49.3101600	100.0000000	100.0000000	100.0000000	0.0000000
22	100.0000000	73.6227500	33.8581100	0.0000000	100.0000000	100.0191000	0.0781715	100.0000000	100.0000000	100.0000000	0.0000001
23	311.9100000	311.4549000	3.1097380	304.0700000	302.8729000	302.0450000	1.4971360	300.0011100	300.0011100	300.0011100	0.0000000
24	100.0000000	102.0606000	24.4169500	0.0000000	329.8444000	292.3530000	82.8574300	100.0000000	100.0000000	100.0000000	0.0000000
25	397.7400000	359.9831000	93.8470700	0.0000000	443.3300000	425.6128000	22.4131600	397.7428700	397.7428700	397.7428700	0.0000000
26	200.0000000	207.8431000	95.5889900	0.0000000	300.0000000	300.0000000	0.0000000	300.0000000	300.0000000	300.0000000	0.0000000
27	389.0800000	389.2237000	0.5055688	387.3200000	388.4065000	388.2803000	3.1282120	373.0700900	373.0700900	373.0700900	0.0000000
28	300.0000000	290.1961000	50.0196000	0.0000000	300.0000000	362.5444000	82.8609900	300.0000000	300.0000000	300.0000000	0.0000009
29	246.4700000	248.4431000	11.8854200	204.9900000	230.3615000	230.4884000	2.2398930	226.9131600	226.9131600	226.9131600	0.0000000
30	504.8400000	540.0845000	119.2817000	400.1000000	217.1836000	229.5842000	32.5351600	202.0146900	202.0146900	202.0146900	0.0000000

Fonte: Próprio Autor

Tabela 36 – Análise Geral (28 funções da CEC-2013 / 10-*D*)

Resultados		CMA-ES	DE/rand/1/bin	CMA-ES-RIS	IPOP-CMA-ES	NBIPPOP-aCMA-ES
L-CMA-DE	Melhor	23	23	12	8	4
	Pior	1	2	10	15	18
	Igual	4	3	6	5	6

Fonte: Próprio Autor

Tabela 37 – Análise Geral (5 funções unimodais da CEC-2013 / 10-*D*)

Resultados		CMA-ES	DE/rand/1/bin	CMA-ES-RIS	IPOP-CMA-ES	NBIPPOP-aCMA-ES
L-CMA-DE	Melhor	1	2	1	0	0
	Pior	1	2	2	3	3
	Igual	3	1	2	2	2

Fonte: Próprio Autor

Tabela 38 – Análise Geral (15 funções multimodais básicas da CEC-2013 / 10-*D*)

Resultados		CMA-ES	DE/rand/1/bin	CMA-ES-RIS	IPOP-CMA-ES	NBIPPOP-aCMA-ES
L-CMA-DE	Melhor	14	14	9	2	1
	Pior	0	0	5	10	10
	Igual	0	1	1	3	4

Fonte: Próprio Autor

Tabela 39 – Análise Geral (8 funções compostas da CEC-2013 / 10-*D*)

Resultados		CMA-ES	DE/rand/1/bin	CMA-ES-RIS	IPOP-CMA-ES	NBIPPOP-aCMA-ES
L-CMA-DE	Melhor	8	7	2	6	3
	Pior	0	0	3	2	5
	Igual	0	1	3	0	0

Fonte: Próprio Autor

Tabela 40 – Análise Geral (29 funções da CEC-2017 / 10-*D*)

Resultados		CMA-ES	DE/rand/1/bin	RB-IPOP-CMA-ES	LSHADE-SPACMA
L-CMA-DE	Melhor	28	19	15	6
	Pior	0	6	10	21
	Igual	1	4	4	2

Fonte: Próprio Autor

Tabela 41 – Análise Geral (2 funções unimodais da CEC-2017 / 10-*D*)

Resultados		CMA-ES	DE/rand/1/bin	RB-IPOP-CMA-ES	LSHADE-SPACMA
L-CMA-DE	Melhor	1	2	0	0
	Pior	0	0	1	1
	Igual	1	0	1	1

Fonte: Próprio Autor

Tabela 42 – Análise Geral (7 funções multimodais simples da CEC-2017 / 10- D)

Resultados		CMA-ES	DE/rand/1/bin	RB-IPOP-CMA-ES	LSHADE-SPACMA
L-CMA-DE	Melhor	7	5	0	0
	Pior	0	1	6	6
	Igual	0	1	1	1

Fonte: Próprio Autor

Tabela 43 – Análise Geral (10 funções híbridas da CEC-2017 / 10- D)

Resultados		CMA-ES	DE/rand/1/bin	RB-IPOP-CMA-ES	LSHADE-SPACMA
L-CMA-DE	Melhor	10	4	1	0
	Pior	0	5	7	10
	Igual	0	1	2	0

Fonte: Próprio Autor

Tabela 44 – Análise Geral (10 funções compostas da CEC-2017 / 10- D)

Resultados		CMA-ES	DE/rand/1/bin	RB-IPOP-CMA-ES	LSHADE-SPACMA
L-CMA-DE	Melhor	10	8	9	6
	Pior	0	0	1	4
	Igual	0	2	0	0

Fonte: Próprio Autor

5 CONCLUSÃO E TRABALHOS FUTUROS

O objetivo principal deste trabalho foi criar um algoritmo evolucionário baseado no *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) que harmonizasse exploração e exploração, para otimizar funções multimodais de um único objetivo, sem causar acréscimo no número de avaliações de função. Para isso foi realizada uma revisão de literatura com o objetivo de auxiliar no entendimento sobre as limitações do CMA-ES, quando aplicado às funções de teste multimodais, ficando esclarecido que o CMA-ES é um ótimo método para realizar exploração, contudo apresenta algumas limitações na otimização de funções multimodais, sendo potencialmente dependente do ponto de inicialização e do tamanho da população. Sendo assim, os esforços foram voltados para conhecer alternativas implementadas ao longo dos anos que procurassem atenuar esta limitação. As abordagens foram agrupadas conforme suas características: baseadas em reinicialização, nichos, inserção de imigrantes e hibridização. Os trabalhos sugerem que agregar um mecanismo de exploração ao CMA-ES, de forma que este possa ser guiado a realizar explorações em regiões promissoras, possibilita a obtenção de melhores resultados em funções com alta multimodalidade.

Diante destas indicações, foram executados testes para compreender o real comportamento da matriz de covariância da população do CMA-ES ao longo das gerações e seu impacto na geração dos indivíduos, ficando demonstrado que a matriz de covariância tende a se tornar esparsa, gerando indivíduos semelhantes entre si. A partir de então, foram implementadas algumas propostas preliminares e executados testes estatísticos, para que fosse possível identificar qual deveria ser a modificação imposta ao CMA-ES, de forma que o seu desempenho em funções multimodais fosse melhorado sem prejuízo do número de avaliações de função. Propostas estas que contemplaram: a manutenção da diversidade populacional através de um outro processo evolucionário que utiliza o CMA-ES baseado na diversidade, que não apresentou diferenças estatísticas ao CMA-ES; a inserção de diversidade através de indivíduos com relevada aptidão, que eram mantidos em uma memória que sofria atualização periódica, a qual apresentou diferença estatística em apenas uma função teste; a inclusão de diversidade à população, através de indivíduos aleatoriamente gerados, o que melhorou estatisticamente a qualidade das soluções obtidas, quando comparadas àquelas geradas pelo CMA-ES original; a inserção de diversidade através de indivíduos gerados com o *Differential Evolution* (DE), que apresentou-se como uma relevante alternativa, possibilitando melhorar os resultados dentre as propostas preliminares já citadas; a utilização da abordagem Método dos Elementos Finitos (MEF) para dividir o espaço de busca, recursivamente, de forma que n CMA-ES fossem executados, o qual indica guiar o CMA-ES para uma região promissora sem desprezar a busca nas demais regiões; por fim, foi elaborada uma abordagem que realizava

a compressão recursiva do espaço de características, delimitado por buscas locais previamente realizadas, o que indicou ser uma proposta promissora, conseguindo atingir o ótimo global em 24 das 28 funções de teste utilizadas.

A partir dos estudos e análises apresentados, foi desenvolvido o algoritmo proposto nesta dissertação, denominado *Local CMA-ES Guided by Differential Evolution* (L-CMA-DE), que agrega a característica de exploração do CMA-ES com a exploração do DE. No L-CMA-DE, são realizadas buscas locais com n CMA-ES independentes que compartilham indivíduos com o DE, de forma que este possa realizar exploração, utilizando informações sobre o ambiente, guiando os CMA-ES para regiões promissoras. Para verificar a qualidade das soluções obtidas pelo L-CMA-DE foram executados testes comparativos, utilizando funções teste de competição da área, com alguns algoritmos que também são baseados no CMA-ES e possuem relevante notoriedade. Como dito no Capítulo 1, as melhorias da nova proposta não poderiam implicar no aumento do número de avaliações de função permitidas. Por este motivo a quantidade máxima de avaliações de função foi a mesma em todos os testes comparativos. Os resultados obtidos indicam que quando aplicado às funções unimodais, o L-CMA-DE apresentou bons resultados, possivelmente devido à facilidade de convergência do próprio CMA-ES nestes tipos de funções. Os piores registros foram obtidos em funções multimodais simples, que são caracterizadas por possuir o ótimo global bem definido (AWAD et al., 2016), onde devido às características do modelo proposto, este não alcançou melhores resultados quando comparado a alguns dos algoritmos testados, muito embora tenha obtido melhores soluções quando comparado aos algoritmos utilizados como base para a construção da proposta (CMA-ES e DE). Espera-se que o L-CMA-DE alcance resultados mais competitivos caso aumente-se a quantidade máxima de avaliações de função permitida. Já nas funções de alta multimodalidade, foi onde o L-CMA-DE obteve seu melhor desempenho, apresentando clara superioridade frente à algoritmos Estado-da-Arte, como por exemplo: CMA-ES, DE e IPOPOP-CMA-ES.

Desta forma, é possível afirmar que o L-CMA-DE consegue harmonizar as características do CMA-ES e do DE (exploração e exploração, respectivamente) de forma a alcançar relevantes resultados em funções teste de alta modalidade, sem o acréscimo na quantidade de avaliações de função permitidas.

Entretanto, ainda são necessários testes para verificar o comportamento do L-CMA-DE com a escalabilidade do espaço de busca. Também se faz necessária uma análise mais aprofundada para identificar o porquê do L-CMA-DE não ter conseguido convergir para o ótimo global, em alguns testes com funções unimodais, além de identificar o motivo do baixo desempenho em funções multimodais simples. Conforme visto na Seção 3.5, o L-CMA-DE possui parâmetros aos quais o modelo não é sensível, devendo portanto ser melhor estudados para verificar o impacto de aplicar-lhes uma autoadaptação. Deve-se também realizar um estudo para verificar a correlação entre todos os parâme-

tro existentes e o modelo proposto, de forma a identificar a possibilidade de controlá-los adaptativamente, suprimindo o ajuste paramétrico. Devido ao comportamento da matriz de covariância, observado nos testes da Seção 3.2, deve-se analisar uma proposta em que um CMA-ES possa ser desprezado, ao passo que os indivíduos de diferentes CMA-ES assemelhem-se em características e aptidão, possibilitando uma economia na quantidade de avaliações de função consumidas, que podem ser utilizadas para dar prosseguimento ao processo evolucionário. Como não há um controle sobre as regiões já exploradas, permitindo que o algoritmo realize buscas em locais exaustivamente examinados, deve-se implementar uma memória para evitar este tipo de comportamento, também com a finalidade de economizar avaliações de função. Uma vez que o objetivo da execução do DE é provocar maior exploração, há um indicativo de que ao se utilizar uma versão deste algoritmo comprovadamente mais exploratória, seja possível obter-se melhores resultados. Por fim, com o objetivo de identificar a melhora nos resultados do L-CMA-DE ao passar das gerações, seria relevante realizar testes aumentando-se a quantidade máxima de avaliações de função permitidas, para analisar a capacidade do algoritmo proposto em continuar melhorando as soluções encontradas, comparado a outros métodos.

REFERÊNCIAS

- AHMAD, I. A. A class of mann—whitney—wilcoxon type statistics. *The American Statistician*, Taylor & Francis Group, v. 50, n. 4, p. 324–327, 1996.
- AUGER, A.; HANSEN, N. A restart CMA evolution strategy with increasing population size. In: IEEE. *Proceedings of the IEEE Congress on Evolutionary Computation*. [S.l.], 2005. v. 2, p. 1769–1776.
- AUGER, A.; HANSEN, N. Tutorial CMA-ES: evolution strategies and covariance matrix adaptation. In: *GECCO (Companion)*. [S.l.: s.n.], 2012. p. 827–848.
- AWAD, N.; ALI, M.; LIANG, J.; QU, B.; SUGANTHAN, P. Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective bound constrained real-parameter numerical optimization. In: *Technical Report*. [S.l.]: Nanyang Technological University Singapore, 2016.
- AZEVEDO, Á. F. *Método dos elementos finitos*. 1^a. ed. [S.l.]: Faculdade de Engenharia da Universidade do Porto, 2003.
- BEYER, H.-G.; ARNOLD, D. V. Qualms regarding the optimality of cumulative path length control in CSA/CMA-evolution strategies. *Evolutionary Computation*, MIT Press, v. 11, n. 1, p. 19–28, 2003.
- BIEDRZYCKI, R. A version of ipop-cma-es algorithm with midpoint for cec 2017 single objective bound constrained problems. In: IEEE. *Evolutionary Computation (CEC), 2017 IEEE Congress on*. [S.l.], 2017. p. 1489–1494.
- CARAFFINI, F.; IACCA, G.; NERI, F.; PICINALI, L.; MININNO, E. A cma-es super-fit scheme for the re-sampled inheritance search. In: IEEE. *Evolutionary Computation (CEC), 2013 IEEE Congress on*. [S.l.], 2013. p. 1123–1130.
- COLETTA, L. F.; HRUSCHKA, E. R.; ACHARYA, A.; GHOSH, J. A differential evolution algorithm to optimise the combination of classifier and cluster ensembles. *IJBIC*, v. 7, n. 2, p. 111–124, 2015.
- DHATT, G.; LEFRANĀ, E.; TOUZOT, G. et al. *Finite element method*. [S.l.]: John Wiley & Sons, 2012.
- EIBEN, A. E.; SMITH, J. E. *Introduction to Evolutionary Computing*. 2^a. ed. [S.l.]: Springer Publishing Company, Incorporated, 2015. ISBN 3662448734, 9783662448731.
- EINSTEIN, A. Geometrie und erfahrung. In: *Geometrie und Erfahrung*. [S.l.]: Springer, 1921. p. 2–20.
- EPITROPAKIS, M. G.; PLAGIANAKOS, V. P.; VRAHATIS, M. N. Balancing the exploration and exploitation capabilities of the differential evolution algorithm. In: IEEE. *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. [S.l.], 2008. p. 2686–2693.

-
- EPITROPAKIS, M. G.; PLAGIANAKOS, V. P.; VRAHATIS, M. N. Finding multiple global optima exploiting differential evolution's niching capability. In: IEEE. *Proceedings of the IEEE Symposium on Differential Evolution (SDE)*. [S.l.], 2011. p. 1–8.
- FRANCO, N. B. *Cálculo numérico*. [S.l.]: Pearson, 2006.
- GOLDBERG, D. *Genetic Algorithms in Search, Optimization, and Machine Learning*. [S.l.]: Addison Wesley, Reading, Massachusetts, 1989.
- HANSEN, N. Benchmarking a bi-population CMA-ES on the BBOB-2009 function testbed. In: ACM. *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. [S.l.], 2009. p. 2389–2396.
- HANSEN, N. Injecting external solutions into cma-es. *arXiv preprint arXiv:1110.4181*, 2011.
- HANSEN, N. *CMA-ES: A Function Value Free Second Order Optimization Method*. 2014.
- HANSEN, N. The CMA evolution strategy: A tutorial. *ArXiv Preprint ArXiv:1604.00772*, 2016.
- HANSEN, N.; ATAMNA, A.; AUGER, A. How to assess step-size adaptation mechanisms in randomised search. In: SPRINGER. *International Conference on Parallel Problem Solving from Nature*. [S.l.], 2014. p. 60–69.
- HANSEN, N.; KERN, S. Evaluating the cma evolution strategy on multimodal test functions. In: SPRINGER. *International Conference on Parallel Problem Solving from Nature*. [S.l.], 2004. p. 282–291.
- HANSEN, N.; MÜLLER, S. D.; KOUMOUTSAKOS, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, MIT Press, v. 11, n. 1, p. 1–18, 2003.
- HANSEN, N.; OSTERMEIER, A. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: IEEE. *Proceedings of the IEEE International Conference on Evolutionary Computation*. [S.l.], 1996. p. 312–317.
- HANSEN, N.; OSTERMEIER, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, MIT Press, v. 9, n. 2, p. 159–195, 2001.
- HANSEN, N.; ROS, R. Benchmarking a weighted negative covariance matrix update on the bbob-2010 noiseless testbed. In: ACM. *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*. [S.l.], 2010. p. 1673–1680.
- HE, X.; ZHOU, Y. Enhancing the performance of differential evolution with covariance matrix self-adaptation. *Applied Soft Computing*, Elsevier, v. 64, p. 227–243, 2018.
- IACCA, G.; MALLIPEDDI, R.; MININNO, E.; NERI, F.; SUGANTHAN, P. N. Super-fit and population size reduction in compact differential evolution. In: IEEE. *Memetic Computing (MC), 2011 IEEE Workshop on*. [S.l.], 2011. p. 1–8.
- IMAN, R. L.; HELTON, J. C. An investigation of uncertainty and sensitivity analysis techniques for computer models. *Risk analysis*, Wiley Online Library, v. 8, n. 1, p. 71–90, 1988.

- JASTREBSKI, G. A.; ARNOLD, D. V. Improving evolution strategies through active covariance matrix adaptation. In: IEEE. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*. [S.l.], 2006. p. 2814–2821.
- KARABOGA, D.; BASTURK, B. On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing*, Elsevier, v. 8, n. 1, p. 687–697, 2008.
- KHAN, N. A parallel implementation of the covariance matrix adaptation evolution strategy. *arXiv preprint arXiv:1805.11201*, 2018.
- KULLBACK, S. *Information theory and statistics*. [S.l.]: Courier Corporation, 1997.
- LI, H.; NANTASENAMAT, C.; MONNOR, T.; ISARANKURA-NA-AYUDHYA, C.; PRACHAYASITTIKUL, V. Genetic algorithm search space splicing particle swarm optimization as general-purpose optimizer. *Chemometrics and Intelligent Laboratory Systems*, v. 128, n. Supplement C, p. 153 – 159, 2013. ISSN 0169-7439. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0169743913001627>>.
- LIANG, J.; QU, B.; SUGANTHAN, P.; HERNÁNDEZ-DÍAZ, A. G. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, v. 201212, p. 3–18, 2013.
- LOSHCHILOV, I. CMA-ES with restarts for solving cec 2013 benchmark problems. In: IEEE. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*. [S.l.], 2013. p. 369–376.
- LOSHCHILOV, I.; SCHOENAUER, M.; SEBAG, M. Black-box optimization benchmarking of nipop-acma-es and nbipop-acma-es on the bbob-2012 noiseless testbed. In: . [S.l.: s.n.], 2012.
- LOSHCHILOV, I.; SCHOENAUER, M.; SEBAG, M.; HANSEN, N. Maximum likelihood-based online adaptation of hyper-parameters in CMA-ES. In: SPRINGER. *Proceedings of the International Conference on Parallel Problem Solving from Nature*. [S.l.], 2014. p. 70–79.
- MCKAY, M. D.; BECKMAN, R. J.; CONOVER, W. J. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, Taylor & Francis, v. 21, n. 2, p. 239–245, 1979.
- MOHAMED, A. W.; HADI, A. A.; FATTOUH, A. M.; JAMBI, K. M. Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems. In: IEEE. *Evolutionary Computation (CEC), 2017 IEEE Congress on*. [S.l.], 2017. p. 145–152.
- MOLINA, D.; PURIS, A.; BELLO, R.; HERRERA, F. Variable mesh optimization for the 2013 CEC special session niching methods for multimodal optimization. In: IEEE. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*. [S.l.], 2013. p. 87–94.
- MÜHLENBEIN, H.; SCHOMISCH, M.; BORN, J. The parallel genetic algorithm as function optimizer. *Parallel computing*, Elsevier, v. 17, n. 6-7, p. 619–632, 1991.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V. et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, v. 12, n. Oct, p. 2825–2830, 2011.

PIMIKOV, A. V. *Ellipse Representing the Confidence Region of a Covariance Matrix*. 2018. Disponível em: <<http://demonstrations.wolfram.com/EllipseRepresentingTheConfidenceRegionOfACovarianceMatrix/>>. Acesso em: 03/07/2018.

PIOTROWSKI, A. P. Review of differential evolution population size. *Swarm and Evolutionary Computation*, Elsevier, v. 32, p. 1–24, 2017.

POTTER, M. A.; JONG, K. A. D. A cooperative coevolutionary approach to function optimization. In: SPRINGER. *Proceedings of the International Conference on Parallel Problem Solving from Nature*. [S.l.], 1994. p. 249–257.

POWELL, M. J. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, Oxford University Press, v. 7, n. 2, p. 155–162, 1964.

PREUSS, M. Niching the CMA-ES via nearest-better clustering. In: ACM. *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*. [S.l.], 2010. p. 1711–1718.

ROS, R.; HANSEN, N. A simple modification in CMA-ES achieving linear time and space complexity. In: SPRINGER. *International Conference on Parallel Problem Solving from Nature*. [S.l.], 2008. p. 296–305.

SCHWEFEL, H.-P. *Evolution and optimum seeking. Sixth-generation computer technology series*. [S.l.]: Wiley, New York, 1995.

SHIR, O. M.; BÄCK, T. Dynamic niching in evolution strategies with covariance matrix adaptation. In: IEEE. *Proceedings of the IEEE Congress on Evolutionary Computation*. [S.l.], 2005. v. 3, p. 2584–2591.

SHIR, O. M.; BÄCK, T. Niching in evolution strategies. In: ACM. *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*. [S.l.], 2005. p. 915–916.

SHIR, O. M.; BÄCK, T. Niche radius adaptation in the CMA-ES niching algorithm. In: SPRINGER. *Proceedings of the Parallel Problems Solving from Nature (PPSN)*. [S.l.], 2006. p. 142–151.

SINGH, M.; PANIGRAHI, B.; ABHYANKAR, A. A hybrid protection scheme to mitigate the effect of distributed generation on relay coordination in distribution system. In: IEEE. *Power and Energy Society General Meeting (PES)*. [S.l.], 2013. p. 1–5.

TSUTSUI, S.; FUJIMOTO, Y.; GHOSH, A. Forking genetic algorithms: GAs with search space division schemes. *Evolutionary computation*, MIT Press, v. 5, n. 1, p. 61–80, 1997.

VENABLES, W.; SMITH, D.; TEAM the R. C. *An Introduction to R: Notes on R: A Programming Environment for Data Analysis and Graphics (Version 3.4.3)*. 2017. Disponível em: <<https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>>. Acesso em: 06/02/2018.

XU, Z.; IIZUKA, H.; YAMAMOTO, M. Attraction basin sphere estimation approach for niching CMA-ES. *Soft Computing*, Springer, v. 21, n. 5, p. 1327–1345, 2017.

XU, Z.; POLOJÄRVI, M.; YAMAMOTO, M.; FURUKAWA, M. Attraction basin estimating GA: An adaptive and efficient technique for multimodal optimization. In: IEEE. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*. [S.l.], 2013. p. 333–340.

YAMAGUCHI, T.; AKIMOTO, Y. Benchmarking the novel CMA-ES restart strategy using the search history on the BBOB noiseless testbed. In: ACM. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. [S.l.], 2017. p. 1780–1787.

ZHOU, H.; QIAO, J. Multiobjective optimal control for wastewater treatment process using adaptive moea/d. *Applied Intelligence*, Springer, v. 49, n. 3, p. 1098–1126, 2019.

ZIELINSKI, K.; WEITKEMPER, P.; LAUR, R.; KAMMEYER, K.-D. Parameter study for differential evolution using a power allocation problem including interference cancellation. In: IEEE. *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. [S.l.], 2006. p. 1857–1864.